## ОСНОВНИ ПОДАТОЦИ ЗА НАСТАВНАТА ПРОГРАМА

| | |
|---|---|
| **Назив на предметот** | **Основи на веб-развој (2)** |
| **Изготвил** | м-р Александар Трајаноски – професор по информатика |
| **Училиште** | ЦОУ “Христо Узунов” – Другово, Кичево |
| **Вид/категорија на наставен предмет** | Слободен изборен предмет |
| **Одделение** | VII, VIII, IX |
| **Број на часови** | 36 |
| **Опрема и средства** | - Аудиовизуелни средства (компјутери со инсталиран Windows 10/11, табла + креда или паметна табла, печатач и други дигитални уреди)<br>- Интернет (линкови/врски, веб-страници)<br>- Уредувачи на код: Visual Studio Code, Sublime Text и слично. |
| **Потребни предзнаења (*препорака*)** | Основи на веб-развој (1) |
| **Наставни материјали** | [OVR2] Наставни материјали 2025 |
| **Контакт** | at146825@schools.mk |

# ПРЕДГОВОР

**Почитувани колеги наставници,**

Со особено задоволство ви ја претставувам наставната програма за „Основи на веб-развој 2“, која претставува логички и содржински надградена целина на претходниот предмет „Основи на веб-развој 1“, во кој се изучуваа темелите на HTML. Овој предмет, кој е дел од изборните програми за проширување и продлабочување на знаењата од задолжителните наставни предмети, има за цел да ги внесе учениците подлабоко во светот на веб-развојот преку изучување на CSS (Cascading Style Sheets) – алатката која ѝ дава стил, структура и визуелен идентитет на секоја веб-страница.

Наставната програма е изработена со внимание на практичната примена и прецизно дефинирани цели. За секој час е подготвено детално сценарио за реализација, што ја прави програмата робусна и функционална, а воедно може да служи и како прирачник при наставната реализација.

И покрај предложената структура, секој наставник има целосна флексибилност во прилагодувањето на часовите, изборот на задачи, начинот на оценување и практичните вежби. Наставната програма треба да се сфати како водич, насока во која може да се движи наставата, оставајќи простор за креативност, експериментирање и прилагодување според потребите на учениците.

Во прилог на оваа програма, достапни се и дополнителни наставни материјали, подготвени да ја олеснат реализацијата на секој час. Материјалите можат да се преземат од Наставни материјали 2025, а доколку наидете на технички проблеми при преземањето, слободно можете да ме контактирате. Стојам на целосно располагање и за сите други прашања, дилеми или нејаснотии поврзани со темите и содржините што се опфатени во програмата.

Свесен сум дека CSS како тема е мошне обемна и дел од неговите аспекти не се вклучени во оваа програма. Сепак, главната цел е да ги насочиме учениците кон самостојно истражување, континуирано учење и развој на вештини за самостојно надоградување. Во денешно време, кога интернетот и вештачката интелигенција се широко достапни, исклучително е важно учениците да научат како да бараат, најдат и применат знаење – а токму оваа програма го поттикнува таквиот пристап.

На сите колеги што ќе ја реализираат оваа програма, ви посакувам успешна, продуктивна и инспиративна работа со учениците. Нека ви биде оваа програма поддршка и мотивација во развивање на следната генерација дигитално писмени и креативни млади луѓе.

Срдечно,

Александар Трајаноски
Контакт: at146825@schools.mk

**Слободен изборен предмет: ОСНОВИ НА ВЕБ-РАЗВОЈ 2**

## ПОВРЗАНОСТ СО НАЦИОНАЛНИТЕ СТАНДАРДИ

Наставната програма вклучува релевантни компетенции од следното подрачје: **IV. ДИГИТАЛНА ПИСМЕНОСТ.**

*Ученикот/ученичката знае и/или умее:*

| | |
|---|---|
| **IV-A.1** | Да ги истражува и споредува можностите на познати и нови дигитални уреди и самостојно да процени, одбере и да ги користи тие кои се најсоодветни за конкретна потреба и ситуација; |
| **IV-A.2** | Да процени кога и на кој начин за решавање на некоја задача/проблем е потребно и ефективно користење на ИКТ, да одбере и инсталира програми кои му/ѝ се потребни, да користи програми за заштита и да реши рутински проблеми во функционирањето на дигиталните уреди и мрежи; |
| **IV-A.3** | Да користи различни начини на организирање и безбедно чување и споделување на содржини на различни уреди и мрежи во дигиталното опкружување; |
| **IV-A.4** | Во соработка со други да анализира проблем, да развие идеја и план за негово истражување и решавање и да испланира кога и за што ќе користи ИКТ; |
| **IV-A.5** | Да определи какви информации му/ѝ се потребни, да најде, избере и преземе дигитални податоци, информации и содржини и да ја процени нивната релевантност во однос на конкретната потреба и веродостојноста на изворот; |
| **IV-A.6** | Да избере и користи различни алатки за обработка на податоци, да ги анализира податоците и да ги претстави на различни начини, почитувајќи ги правилата за користење; |
| **IV-A.7** | Да одбере и користи соодветни ИКТ-алатки за комуникација, безбедно да сподели информации, да контактира и да соработува со други на онлајн проекти, во социјални активности или за лични потреби; |
| **IV-A.8** | На безбеден и одговорен начин да ги користи дигиталните содржини, образовните и социјалните мрежи и дигиталните облаци; |
| **IV-A.11** | Да планира и да развива секвенци од јасни инструкции за изведување конкретна задача и да ги прикаже како програмски алгоритам; |
| **IV-A.12** | Да истражува можности за користење на различни модели и симулации, комбинирање статични и динамички претставувања, звук, текст и слики за да модифицира или создаде едноставни креативни мултимедиумски продукти со конкретна намена и за определена публика; |
| **IV-A.13** | Да дефинира критериуми за квалитетот на дигиталните продукти и решенија, вклучувајќи ги иновативноста и корисноста. |

*Ученикот/ученичката разбира и прифаќа дека:*

| | |
|---|---|
| **IV-Б.1** | Дигиталната писменост е неопходна за секојдневното живеење – го олеснува учењето, животот и работата, придонесува за проширување на комуникацијата, за креативноста и иновативноста, нуди разни можности за забава; |
| **IV-Б.3** | Потенцијалите на ИКТ ќе се зголемуваат и треба да се следат и користат, но и дека треба да се има критичен однос кон веродостојноста, доверливоста и влијанието на податоците и информациите кои се достапни преку дигиталните уреди; |
| **IV-Б.5** | Информациите достапни во дигиталниот простор треба да се користат етички, според дефинирани правила, и за добро на луѓето; |
| **IV-Б.6** | Мора да се почитува правото на интелектуална сопственост на продуктите достапни на дигиталните мрежи; |
| **IV-Б.7** | Неумереното и во несоодветна положба (неергономски) користење на дигиталните технологии може негативно да влијае на здравјето, на личниот и социјалниот живот, а несоодветното складирање на дигиталниот отпад неповолно влијае врз животната средина |

Наставната програма вклучува дополнителни релевантни компетенции и од следните подрачја на Националните стандарди: **VII. ТЕХНИКА, ТЕХНОЛОГИЈА И ПРЕТПРИЕМНИШТВО.**

*Ученикот/ученичката знае и/или умее:*

| | |
|---|---|
| **VII-А.4** | Да генерира идеи и осмислува активности кои водат до продукти и/или услуги; |
| **VII-А.9** | Активно да учествува во тимска работа според претходно усвоени правила и со доследно почитување на улогата и придонесот на сите членови на тимот; |
| **VII-А.11** | Да донесе одлука за натамошно школување врз основа на сопствените интереси, способности и можности, земајќи ги предвид потребите на пазарот на трудот; |

*Ученикот/ученичката разбира и прифаќа дека:*

| | |
|---|---|
| **VII-А.4** | Успешните идеи кои водат кон лични, социјални и финансиски придобивки се резултат на креативност, иницијативност, посветеност и истрајност; |
| **VII-А.9** | Иницијативата е битен услов за внесување промени во личниот живот и животот во заедницата, а успешноста на промените е поврзана со соочување со предизвици и/или преземање ризик; |

## РЕЗУЛТАТИ ОД УЧЕЊЕ

**Тема 1. Теоретски вовед во CSS, неговите почетоци, развој и примена (4 часа)**

***Знаења/вештини:***

- Прави разлика меѓу HTML5 (*HyperText Markup Language 5*) и CSS3 (*Cascading Style Sheets 3*) и објаснува како заедно функционираат во структурата и изгледот на веб-страниците.
- Го препознава развојот на CSS и неговите верзии (од CSS1 до CSS3 и понатаму).
- Го објаснува значењето и улогата на CSS во изработката на веб-страници.
- Ги знае трите начини за вчитување и внесување на CSS код во HTML документ;

***Ставови/вредности:***

- Ја прифаќа важноста на добрата визуелна презентација и нејзиното влијание врз корисничкото искуство при користење веб-страни.
- Има позитивен став кон учењето на програмски јазици како и јазици за развој на веб-страници.
- Подготвен е да презема активности кои вклучуваат самостојно и тимско изработување на дизајни користејќи CSS.
- Покажува интерес и љубопитност за веб-технологии, вклучително и нивната еволуција.
- Разбира дека CSS како и останатите програмски јазици е јазик кој постојано се развива и бара следење на нови трендови и техники.

| ***Содржини (и поими) и број на часови*** | ***Примери на активности:*** |
|---|---|
| • **Што претставува CSS и за што се користи?**<br>(HTML. CSS. Cascading Style Sheets. Визуелен приказ. Стил) | **Вовед**<br>Наставникот ги запознава учениците со целите, темите и содржините на наставната програма. Преку усно излагање и кратка презентација ги запознава учениците со CSS3 (Cascading Style Sheets Level 3): “CSS претставува јазик за стилско уредување (т.е. визуелен приказ) и се користи за дефинирање на изгледот и распоредот на HTML елементите во веб-страницата. |

| | |
|---|---|
| Број на часови: 2 (1, 2) | Со CSS, можеме да ја одделиме структурата, односно “скелетот” на веб-страницата (кој се дефинира со HTML) од нејзината визуелна презентација, со што се овозможува почист, поодржлив и пофлексибилен код.”<br><br>**Почетоци и развој**<br><br>CSS е развиен од W3C (World Wide Web Consortium), а првата верзија – CSS1, официјално е објавена во 1996 година. Причината за неговото создавање била потребата од стандардизиран начин за стилско уредување на веб-страници.<br><br><u>CSS се користи за:</u><br>○ Поставување на бои, фонтови, големини, маргини и други стилски својства на елементите.<br>○ Респонзивен дизајн, кој овозможува веб-страниците да се прилагодуваат на различни уреди (мобилни, таблети, десктоп).<br>○ Анимации и транзиции, со цел за додавање интерактивни ефекти.<br>○ Изработка на флексибилни распореди со помош на flexbox и grid.<br>○ Одделување на дизајнот од содржината, што ја олеснува одржливоста и повторната употреба на стилови.<br><br>1) На следниот линк: https://www.w3schools.com/css/css_intro.asp (***One HTML Page - Multiple Styles!***) има одличен пример за презентирање на тоа како една иста структура или “скелет” направена во HTML може да добие различни **визуелни прикази** или **стилови** изработени со CSS.<br><br>2) Се презентираат двата ***html*** документи од наставните материјали (1. Вовед) каде што првиот ***primer1.html*** е структура претставена само со HTML, додека во вториот ***primer2.html*** има додадено стилско уредување на содржината со помош на CSS3. |

Во вториот пример* направени се следните промени:

- Бојата на позадината сменета во ***lightgrey;***
- Текстот во документот е порамнет на средина;
- Направена е промена во стилот, односно типот на фонтот (од **serif** "декоративен" фонт во **sans-serif** т.е. од стандардниот Times New Roman во Arial);
- Променета е бојата на текстот во насловот **<h1>** и параграфот **<p>;**
- Големината на текстот во параграфот е променета во **22px,** наместо стандардната големина од **16px**.

* Во овој момент не се очекува учениците да знаат што означува и како функционира кодот во делот <style>, туку примерот служи за нивно запознавање со материјата т.е. приближно да се добие претстава како изгледа, што може да се направи со CSS и која е неговата примена.

3) Се презентира и следната табела со споредбените карактеристики:

**Кратка споредба помеѓу HTML и CSS**

| **Карактеристика** | **HTML** | **CSS** |
|---|---|---|
| **Функција** | Структура на веб-страницата (скелетот) | Изгледот т.е. стилското уредување на веб-страницата |
| **Тип на јазик** | Јазик за означување (**Markup Language**) | Јазик за стилско уредување (**Stylesheet Language**) |
| **Примена** | Ги дефинира елементите: наслови, пасуси, линкови итн. | Ги уредува: боите, фонтот, анимациите, распоредот и сл. |
| **Зависност** | Може да функционира самостојно | Зависи од HTML, го уредува стилот на однапред дефинираните ознаки. |

4) Учениците даваат свои размислувања на следните прашања:

- Дали една веб-страница може нормално да функционира доколку е изработена само во HTML, и обратно, дали па една веб-страница може да се изработи само со користење на CSS?

- Кои карактеристики ќе ги има веб-страницата која ќе биде изработена само со HTML и каква ќе биде финалната визуелна презентација?

  ! Кај одговорите на ова прашање, очекувано е учениците да имаат "негативен" поглед во однос на финалниот изглед кај статичката веб-страницата изработена само со HTML, токму поради нејзиниот "здодевен" дизајн. Но, тука наставникот треба да ги нагласи и позитивните страни, а тие се дека оваа "здодевна" веб-страница може да има високи перформанси во однос на нејзината брзина со која се вчитува во пребарувачот. Поради тоа, оваа веб-страница без проблем и со голема брзина целосно може да се прикаже во пребарувачот, па дури и кога би го користеле најбавниот Интернет во светот. 😉

**Се донесува следниот заклучок:**

- HTML ја создава основната содржина **(Што гледаме)**;
- CSS ја уредува и "разубавува" таа содржина **(Како изгледа)**;
- JavaScript* додава интерактивност и динамично однесување на елементите **(Како реагира**: кликање на копче, отворање мени, валидација на форма итн.**)**

Овие три технологии заедно ја формираат основата на секоја модерна веб-страница: HTML за структура, CSS за изглед и JavaScript за функционалност/интерактивност.

- **Поставување на работна околина и начини на вметнување на CSS**

  (Работна околина. Папка. Уредувач на код. CSS датотека.)

  Број на часови: 2 (3, 4)

Препорачливо е учениците да се организираат да работат во парови или тандем, односно во групи, во зависност од опременоста на кабинетот по информатика. На својот компјутер учениците креираат папка во која ќе ги чуваат сите изработки, задачи, вежби и проекти од наредните часови. (*Примери за именување на папка:* ***OVR2-2025, OVR2 VIII-1*** *и сл.*)

Препорака за уредувач на код е **Sublime Text**, поради тоа што е бесплатен, има едноставен интерфејс и нема високи хардверски побарувања. (Можат да се користат и други уредувачи на код, како на пр: *Visual Studio Code, Notepad++, Atom* итн.)

Се стартува уредувачот на код **Sublime Text** и од менито ***File*** се избира опцијата ***Open Folder...*** Во ново отворениот прозорец се оди до локацијата од претходно креираната папка (пр. ***OVR2-2025***), се селектира и се притиска на копчето ***Select Folder***. За да се креира нова датотека, од менито **File** се избира опција ***New File***. Со командата ***File -> Save*** или комбинацијата од тастери ***Ctrl + S,*** се зачувува новокреираната датотека под името *index.html*. **(Ова е важно, бидејќи со додавање на екстензијата .html му се дава до знаење на уредувачот на код за каков тип на датотека станува збор, со цел внесениот код автоматски да биде форматиран согласно дефинираните поставки зададени во уредувачот за соодветниот програмски јазик)**

Во датотеката index.html со внесување на текстот "*html*" и притискање на копчето **tab** од тастатурата се прави **автоматско дополнување** (*autocomplete*) на документот со основната HTML структура [*html* + press **tab**]:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title></title>
</head>
<body>

</body>
</html>
```

**Форми и начини на вметнување на CSS**

CSS може да се користи на три начини:

1. ***Inline*** – Директно во HTML елементот (пр. style="color:red;");
2. ***Internal*** – Во <style> ознаката во самата HTML датотека;
3. ***External*** – Преку поврзување со надворешна .css датотека **(најпрепорачан начин)**.

Од наставните материјали во "2. Работна околина" се презентираат трите датотеки именувани како ***inline.html***, ***internal.html*** и ***external.html,*** кои во пребарувачот (*Google Chrome, Microsoft Edge, Mozilla Firefox* итн.) го даваат истиот визуелен резултат, меѓутоа начинот со кој тоа е постигнато е различен. На секоја страница од менито со десен клик се избира опцијата ***View Page Source*** или директно се користи кратенката ***Ctrl + U*** со цел засебно да се разгледа кодот на секој .html документ. Притоа се согледуваат разликите во уредувањето на HTML документите со помош на CSS, на кои начини тоа е направено и се дискутира за предностите и недостатоците на секој од нив.

**Недостатоци на Inline CSS**

- Неможност за повторна употреба (Стиловите мора да се повторуваат за секој елемент поединечно);
- Тежок за одржување (Ако треба да се промени стил, мора да се најдат и изменат сите елементи рачно);
- Нарушена читливост (HTML кодот станува долг и неуреден, бидејќи стиловите се мешаат со структурата).

**Недостатоци на Internal CSS**

- Не може да се искористи на повеќе страници одеднаш (Ако веб-страницата има повеќе HTML-датотеки, истиот стил мора да се копира во секоја);
- Поголем код по страница (Секој HTML документ со вграден <style> блок станува подолг и потежок);
- Потенцијално мешање со други стилови (Ако се комбинира со надворешни или inline стилови, може да дојде до појава на конфликти и неочекувано однесување на елементите);

**Предности на External CSS во однос на претходните два начини**

- Повторна употреба (Една CSS датотека може да се користи за повеќе HTML страници);
- Лесно одржување (Промените се прават на едно место во .css датотека и автоматски важат за сите поврзани страници);
- Подобра организација на код (Одделувањето на HTML (структура) и CSS (дизајн) го прави кодот почист,

| | |
|---|---|
| | поразбирлив и полесен за навигација);<br>○ Побрзо вчитување (кеширање);<br>○ Професионален пристап (External CSS е индустриски стандард и се користи во сите професионални веб-проекти);<br>○ Поддржува напредни функции.<br><br>**Се донесува следниот заклучок:**<br><br>**Inline** и **Internal CSS** можат да бидат корисни за брзи или мали проекти, но во реални веб-апликации и поголеми веб-страници најдобра пракса е користење на **External CSS**, бидејќи овозможува чист код, брзо и полесно одржување, како и подобри перформанси.<br><br>Во воведните часови препорачливо е учениците да користат **Inline CSS** во ознака **<style>** во самиот HTML документ, односно во периодот додека се запознаваат со синтаксата на CSS. Потоа, во понатамошните задачи, вежби или проекти кои се предвидени во оваа наставна програма, може да се премине на **External CSS** или методот во кој структурата се наоѓа во еден, а дизајнот во друг документ. |

**Тема 2. Основи на CSS и практична примена (25 часа)**

***Знаења/вештини:***

- Применува основни CSS правила за стилизирање текст, бои, маргини, рамки и фонтови.
- Креира едноставна веб-страница користејќи HTML и CSS.
- Изработува мини-проекти со CSS со почитување на принципите на визуелна хиерархија и естетика.
- Работи одговорно и самостојно, но и во тим, почитувајќи туѓи идеи и дизајни.
- Го почитува авторството и ги користи лиценцирани извори за инспирација и учење.
- Развива критичко мислење за визуелниот изглед и корисничкото искуство на веб-страници.

***Ставови/вредности:***

- Смета дека учењето на јазикот CSS е значајно за развој на дигитални вештини и подготовка за идните технолошки предизвици.
- Ја прифаќа важноста на добрата визуелна презентација и нејзиното влијание врз корисничкото искуство при користење веб-страни.
- Подготвен е да презема активности кои вклучуваат самостојно и тимско изработување на дизајни користејќи CSS.
- Ја осудува злоупотребата на технологија и копирањето на туѓа работа без дозвола или цитирање извор.
- Покажува интерес и љубопитност за веб-технологии, вклучително и нивната еволуција.
- Верува дека секој ученик, без разлика на претходни знаења, може да ги совлада основите на CSS со соодветна поддршка.
- Го поддржува вклучувањето на практична работа во наставата преку проекти, веб-страници и интерактивни примери.
- Разбира дека CSS како и останатите програмски јазици е јазик кој постојано се развива и бара следење на нови трендови и техники.
- Се залага за создавање инклузивна и охрабрувачка средина за сите ученици кои сакаат да се изразат преку дизајн и технологија.

| *Содржини (и поими) и број на часови* | *Примери на активности:* |
|---|---|
| • **Основна синтакса на CSS**<br><br>(Селектор. Атрибут. Вредност. Блок за декларација. CSS правило.)<br><br>Број на часови: 2 (5, 6) | Целите на овој час се учениците да ја разберат структурата и основната синтакса на CSS, да разликуваат што се тоа селектор, атрибут (својство) и негова вредност и да можат успешно да применат и напишат валидно CSS правило со точна синтакса.<br><br>Се презентира график или слика со структура од CSS код:<br><br>`p {`<br>`  color: red;`<br>`  font-size: 20px;`<br>`}`<br><br>**Опис на структурата:**<br><br>1. **p** => **Селектор** (Го одредуваме HTML елементот врз кој сакаме да примениме стил – пр. параграф);<br>2. color и font-size => **Атрибут** (Го одредуваме атрибутот на HTML елементот што сакаме да го промениме, во конкретниот случај БОЈА и ГОЛЕМИНА НА ФОНТ на текстот во параграфот);<br>3. **red** и **20px** => **Вредност** (Се дефинира вредноста што сакаме да ја доделиме на атрибутот– црвена боја на текстот и големина од 20 пиксели);<br>4. Просторот помеѓу заградите, односно местото каде што се пишува CSS кодот или CSS правилото { ... } се нарекува **блок за декларација/декларациски блок.**<br>5. ВАЖНО! После секој **атрибут** се ставаат **две точки ( : ).** После секоја **вредност** се става **точка и запирка ( ; ).** Така, во еден декларациски блок можеме да имаме **повеќе CSS правила, одделени со точка и запирка**. Препорачливо е секое ново CSS правило да започнува во нов ред.<br><br>**Задача за учениците:**<br><br>1) Се стартува *Sublime Text* и на секој компјутер со отвора соодветната папка креирана од тимот или групата на ученици. Се избира HTML документот именуван index.html креиран на |

| | |
|---|---|
| | претходните часови. Во делот **<head>** се креира нова ознака **<style> … </style>** во кој ќе се внесува CSS кодот **(Internal CSS)**.<br><br>2) Креирај наслов **<h1>***Добредојде во светот на CSS***</h1>** и параграф ** ***Ова е текст со црвена боја и порамнет на средина*.** .** Во делот **<style>** со валидни CSS правила примени ги следните својства (*color*, *font-size*, *text-align*) на соодветните елементи:<br>○ Наслов **<h1>** сина боја, порамнет на средина;<br>○ Параграф ** ** црвена боја, големина на фонт од 18p и порамнет на средина.<br><br>3) Учениците ги увежбуваат стекнатите вештини, додаваат нови елементи во документот и ги уредуваат по желба.<br><br>4) Се задава квиз со прашања за синтаксата на CSS (Наставен материјал од **3. Синтакса: [Квиз] Основна синтакса на CSS**). Квизот може да се даде испечатен или да се сподели електронски на секој ученик или тим/група. Верзијата дадена во ***.docx*** ги содржи и точните одговори, додека ***.pdf*** документот е без решенија, подготвен за печатење.<br>Прашањата од квизот (на англиски јазик) можат да се пронајдат на следниот линк: https://www.w3schools.com/css/css_syntax.asp на дното од страницата во делот ***Exercise***. |
| • **Стилизирање на текст**<br><br>(Фонт. Боја. Големина. Порамнување.)<br><br>Број на часови: 3 (7 – 9) | Целите на овие часови се учениците да го разберат и користат CSS во постапките наменети за стилизирање на текстот, каде се вклучени изборот на фонтот, големината, бојата и неговото порамнување во однос на страницата.<br><br>**Атрибути за стилизирање на текст – прв дел:** *color, font-family, font-size, text-align*.<br><br>1. Атрибутот ***color*** ја одредува **бојата** на текстот. При доделување на вредност може да се користи името на бојата *(на пр. **red**),* хексадецимален број *(**#ff0000**)* или RGB *(**rgb(255, 0, 0)**)*. Сите три начини како резултат ја даваат црвената боја за боја на текстот во избраниот елемент. |

Пример за стилизирање на текст со **темно-портокалова** боја:

- ***color:*** *DarkOrange;*
- ***color****: #FF8C00;*
- ***color:*** *rgb(255, 140, 0);*

На сајтот https://htmlcolorcodes.com/ има огромен број на палети на бои групирани во најразлични категории заедно со нивните имиња, хексадецимални кодови и RGB вредности.

2. Изборот на соодветен фонт има големо влијание врз тоа како посетителот ја доживува нашата веб-страница. Со атрибутот ***font-family*** се одредува **типот на фонтот** кој ќе се користи за текстот во избраниот елемент.

Во CSS постојат пет генерички семејства на фонтови:

- **Serif** фонтови – имаат мали завршетоци (украсни додатоци) на рабовите од секоја буква. Создаваат чувство на формалност и елеганција.
- **Sans-serif** фонтови – имаат чисти линии (без "украси"). Создаваат модерен и минималистички изглед.
- **Monospace** фонтови – сите букви имаат иста, фиксна ширина. Создаваат "машински" изглед.
- **Cursive** фонтови – буквите имаа сврзни линии и имитираат човечки ракопис.
- **Fantasy** фонтови – пред сè декоративни фонтови. Се користат за креативни и разиграни дизајнерски проекти.

Разлика помеѓу **Serif** и **Sans-serif** фонт:


**Забелешка:** На компјутерски екрани, **sans-serif** фонтовите се сметаат како полесни за читање од **serif** фонтовите.

Табеларен приказ на генерички семејства и примери од имиња на фонтови од соодветната група:

| Генеричко семејство на фонт | Име на фонт |
|---|---|
| **serif** | Times New Roman<br>Georgia<br>Garamond |
| **sans-serif** | Arial<br>Verdana<br>Helvetica |
| **monospace** | Courier New<br>Lucida Console |
| **cursive** | Comic Sans<br>Brush Script MT<br>Lucida Handwriting |
| **fantasy** | COPPERPLATE GOTHIC<br>Papyrus |

Пример за CSS правило за дефинирање на фонт:

```
p {
    font-family: Helvetica, Arial, sans-serif;
  }
```

**Совет:** Атрибутот **font-family** во вредност треба да содржи неколку имиња на фонтови како "резерва" (најмалку две) со цел да се обезбеди максимална компатибилност помеѓу различни прелистувачи и оперативни системи. Се започнува со фонтот со кој што сакаме да се прикаже текстот (Helvetica), па потоа Arial, и на крај да заврши со соодветно генеричко семејство **sans-serif** (Ако се случи претходните фонтови да не се достапни, прелистувачот ќе избере сличен фонт од тоа семејство, т.е sans-serif). Имињата на фонтовите треба да бидат разделени со запирка.

! Доколку името на фонтот се состои од повеќе зборови (пр. Times New Roman), тој фонт мора да биде ставен во **наводници** (" ") во декларацијата на CSS правилото.

```
h1 {
    font-family: "Times New Roman", Times, serif;
  }
```

или

```
a {
    font-family: "Lucida Console", "Courier New", monospace;
  }
```

**Задача за учениците:**

1) Се стартува *Sublime Text* и на секој компјутер со отвора соодветната папка креирана од тимот или групата на ученици. Се креира нов фајл и се зачувува под името ***text.html*** **(**Документот е достапен во наставните материјали: **4. Стилизирање на текст)**. Се користи постапката за автоматско дополнување (Се внесува текстот HTML и се притиска на копчето **tab**). Во делот **<head>** се креира нова ознака **<style> … </style>** во кој ќе се внесуваат CSS правилата. Во делот **<body> … </body>** се креира html структура која ќе ги содржи следните елементи: *h1, h2, h3, p, ul, li, hr*. (За поефикасно пишување код и тука се користи постапката за автоматско дополнува, односно се впишува името на елементот, а веднаш потоа се притиска на копчето **tab**).

2) Откако ќе се заврши html структурата, учениците ги стилизираат елементите по нивна желба користејќи ги атрубутите ***color*** и ***font-family***.

Се користи https://htmlcolorcodes.com/ за избор на бои со соодветното име или хексадецимален код. Исто така, може да се користи и следната листа на “Веб-безбедни” фонтови, т.е. фонтови со

најголеми шанси дека со сигурност и без проблем ќе бидат прикажани во пребарувачот: https://www.w3schools.com/cssref/css_websafe_fonts.php.

3) Учениците истражуваат, експериментираат, додаваат свои елементи и ја уредуваат содржината на страницата по нивна желба.

3. Атрибутот ***font-size*** ја одредуба **големината** на текстот во елементот. Мерните единици т.е. вредностите кои ги прима овој атрибут се: **px, em, rem, %, vw.** Во вежбите и задачите од оваа наставна програма ќе се користи мерната единица ***px*** *(pixels)* за стилизирање на големината на текстот.

```
p {
    font-size: 22px;
  }
```

4. Атрибутот ***text-align*** се користи за порамнување на текстот во елементот. Овој атрибут ги прима следните вредност: ***left*** *(лево)*, ***right*** *(десно)*, ***center*** *(средина)* и ***justify*** *(порамнување од двете страни)*.

```
p {
    text-align: center;
  }
```

**Задача за учениците:**

1) Се користи датотеката именувана како ***text.html***. Се даваат насоки за уредување на текстот преку менување на големината и негово порамнување. И во оваа вежба се дава слобода на учениците да истражуваат и да ја уредуваат содржината на страницата по нивна желба.

ВАЖНО! Способноста за управување со големината на текстот е многу корисна во веб-дизајнот. Сепак, не треба да се користи прилагодување на големината на фонтот со цел да се направи параграфите да изгледаат како наслови, или насловите да изгледаат како параграфи. Ако не се зададе големина на фонтот, стандардната големина за обичен текст, на пр. параграф *p*, е **16px** (16px = 1em); наслов *h1*, е **32 px** (32px = 2em); наслов *h2*, е **24px** (24px = 1.5em); итн.

| | |
|---|---|
| | 4) Се задава квиз со прашања за стилизирање на текст во CSS (Наставен материјал од **4. Стилизирање на текст: [Квиз] Стилизирање на текст во CSS**). Квизот може да се даде испечатен или да се сподели електронски на секој ученик или тим/група. Верзијата дадена во ***.docx*** ги содржи точните одговори, додека ***.pdf*** документот е без решенија, подготвен за печатење. |
| • **Вчитување на фонт од надворешни извори (Google Fonts)**<br><br>(Онлајн библиотека. Типографија. Google Fonts.)<br><br>Број на часови: 1 (10) | **Google Fonts** е бесплатна онлајн библиотека со голем избор на фонтови што можат лесно да се вметнат и користат во веб-страници преку HTML и CSS. Овозможува модерен и креативен изглед на текстот, без потреба корисникот да ги има фонтовите инсталирани на својот уред. Фонтовите се оптимизирани за брзо вчитување и компатибилност со сите модерни прелистувачи.<br><br>Наставникот ја презентира содржината на бесплатната онлајн библиотека за фонтови https://fonts.google.com/. Во делот од левата страна преку формата за внесување текст **“preview”,** можеме да добиеме преглед на внесениот текст низ огромниот број фонтовите кои стојат на располагање поделени во најразлични категории. Се презентира и постапката за вчитување на фонт во **html** документ и потоа како се користи со **CSS**.<br><br>! **Во делот “Language” има опција за внесување јазик, на пр. ако се внесе македонски јазик (Macedonian) ќе се излистаат фонтовите кои имаат кирилична поддршка.**<br><br>**Задача за учениците:**<br><br>1) Се креира нова датотека и се зачувува по името ***cool-fonts.html***. Се прави автоматско пополнување со ***html*** структура и во **<*head*>** се додава нова ознака **<*style*> … </*style*>**.<br><br>2) Се посетува веб-страницата: https://fonts.google.com. Се избира фонт (на пр. *Roboto, Lobster, Open Sans* итн.) и се притиска на копчето ***Get fonts***, а потоа на **<> *Get embed code***. Се копира кодот кој се наоѓа во табот **web** под опцијата **<link>**. (!Пред да се копира линкот има можност за доуредување на фонтот со дополнителни стилови на копчето ↓ **Change styles**). Копираниот линк се “залепува” во **html** документот веднаш над ознаката **<style> … </style>** во делот **<head>**.<br><br>3) Се креира наслов **<h1>** со произволна содржина (пр. *“Ова е наслов напишан со фонт* |

*преземен од Google Fonts”* и сл.) и два параграфи со **привремен текст***. Таков текст се прави со внесување на зборот *Lorem* и пристискање на копчето **tab** во местото каде што сакаме да се појави текстот [** ***Lorem* + press ***tab***** **].

***! *Lorem ipsum* е стандарден привремен текст што се користи во графички дизајн, веб-дизајн и издаваштво како замена за реална содржина. Потекнува од латински текстови и се користи за визуелно претставување на изгледот на страници, фонтови или дизајн, без да го одвлекува вниманието со вистински зборови. Служи за да се оцени типографијата, распоредот и визуелната рамнотежа на дизајнот пред да се внесе вистински текст.**

4) Во **<style> … </style>** се задаваат CSS правилата за стилизирање на текстот:

- За вредност на атрибутот **font-family** се избира името на фонтот *Roboto* проследен со генеричкото семејство на фонтови на кој припаѓа *sans-serif*;
- Текстот се порамнува на средина со задавање вредност *center* на атрибутот **text-align**;

**Совет!** Ако имаме случај каде што повеќе елементи споделуваат исти CSS правила, препорачливо е да се организирани во еден заеднички декларациски блок. Тоа може да се постигне ако во делот за селектор, одделени со запирка, се наведуваат елементите кои споделуваат исти вредности.

```
h1, p {
    font-family: "Roboto", sans-serif;
    text-align: center;
  }
```

**Атрибути за стилизирање на текст – втор дел:** *font-style, font-weight, text-decoration, line-height.*

Постојат и други атрибути што се користат за дополнително стилизирање на текстот, т.е да се нагласи, истакне или промени визуелниот изглед на текстот во веб-страницата. Тоа се:

- ***Font-style*** го одредува **стилот** на фонтот, како на пр. *italic* (искосено).
- ***Font-weight*** ја одредува дебелината на буквите, прима вредности од 100, 200, 300 па сè до 900.
- Со ***text-decoration*** се прави **подвлечен** (*underline*) или **прецртан** (*line-through*) текст.

- ***Line-height*** дефинира **проред**, односно растојанието помеѓу редовите на текстот (прима вредности од број, процент и px)

**Задача за учениците:**

1) Во документот ***cool-fonts.html*** насловот се стилизира со дебелина на буквите 100, големина од 56px и стил со искосени букви.

2) Текстот во параграфите се стилизира со дебелина на буквите 200, големина од 18px, а проредот добива вредност од 1.8

```
h1 {
        font-weight: 100;
        font-size: 56px;
         font-style: italic;
    }

 p {
        font-weight: 200;
        font-size: 18px;
         line-height: 1.8;
    }
```

Наставникот пожелно е да издвои и да им прикаже на учениците одредени недостатоци при користењето на фонтови од *Google Fonts*, како што се на пример:

1. **Зависност од интернет конекција** - *Google Fonts* се вчитуваат од надворешен сервер ***(fonts.googleapis.com),*** па без интернет или при слаб сигнал фонтовите можеби нема да се прикажат правилно.

2. **Забавено вчитување на страницата** - Додавањето повеќе фонтови или варијации (различни дебелини, стилови) може да ја забави брзината на вчитување, особено на мобилни уреди или бавен интернет.

| | |
|---|---|
| | 3. **Ограничено визуелно прилагодување** - Иако изборот е голем, некои фонтови не поддржуваат специфични јазични знаци (како кирилица, грчки, арапски), или немаат доволно стилови (*italic, light, extra-bold* итн.). |
| • **Работа со CSS позадини**<br><br>(Позадина. Боја. Слика. Текстура.)<br><br>Број на часови: 3 (11 – 13) | За поставување и стилизирање на позадината на елементите во веб-страницата се користат атрибутите ***background-***. Со нив може да се зададе боја, слика или текстура на позадината на елементот. Исто така, со нив може да се постават различни големини и вредности на внесените слики и текстури, како и број на нивно повторувања во елементот.<br><br>Најчесто користени атрибути се:<br><br>○ ***background-color*** – Се задава боја на позадина во елементот. За вредност се внесува: име на боја, хексадецимален број на боја или **rgb** вредност.<br><br>○ ***background-image*** – Како позадина на елементот, наместо боја одредува **слика** или **текстура**. За вредност се внесува **адресата/локацијата** на избраната слика или текстура.<br><br>○ ***background-repeat*** – Во која насока ќе се повторува сликата или текстурата. Ги содржи следните вредности: ***repeat-x*** *(повторување по хоризонтала),* ***repeat-y*** *(повторување по вертикала),* ***no-repeat*** *(без повторување).* *ВАЖНО!* *Доколку се внесува димензиски помала слика или текстура вчитана преку* ***background-imade*** таа стандардно ќе се повторува во двете насоки и хоризонтално и вертикално. Во тој случај се користи вредноста ***no-repeat,*** за да нема повторување на сликата/текстурата.<br><br>○ ***background-size*** – се користи за дефинирање на големината на сликата/текстурата. Прима вредности во проценти. (На пр. 100% значи дека сликата/текстурата ќе биде развлечена на целиот екран)<br><br>○ ***background-position*** – Ја одредува позицијата на сликата/текстурата во однос на границите на елементот. Ги содржи следните вредности: *left top, left center, left bottom, right top, right center, right bottom, center top, center center, center bottom.* (Самиот опис покажува каде ќе биде позиционирана сликата/текстурата. На пример: ***right top*** ќе ја позиционира во горниот десен агол од рамката на елементот). |

**ЗАДАЧА ЗА УЧЕНИЦИТЕ:**

1) Се креира нов документ именуван ***pozadini.html*** (или ***backgrounds.html***). Во новиот документ се копира содржината од документот ***cool-fonts.html***.

2) Во ознаките **<h1>** и **<p>** во делот **<style> ... </style>** се додаваат CSS правила за додавање боја на позадината во текстот. Се користи веб-страницата https://htmlcolorcodes.com/ за избор на бои.

```
h1 {
        background-color: #abebc6;
    }

p   {
        background-color: #fdebd0;
    }
```

3) Се креира нова папка под името **img** и се симнуваат неколку позадини. Потоа се вчитуваат со користење CSS правило во ознаката **<body>**.

```
body {
        background-image: url("img/floral.webp");
        background-repeat: no-repeat;
        background-size: 100%;
    }
```

4) Се посетува веб-страницата https://www.transparenttextures.com/ за преземање на транспарентни текстури кои ќе се користат за позадини. Овие текстури одат во комбинација со **background-color**, односно самата текстура може да се "обои" со боја по наша желба.

Друг алтернативен извор на текстури има на https://www.magicpattern.design/tools/css-backgrounds. (Опциите ***Opacity*** и ***Spacing*** служат за директно уредување на текстурата пред

| | |
|---|---|
| | нејзино преземање на компјутер)<br><br>`body {`<br>`  background-image: url("img/binding-light.png");`<br>`  background-color: black;`<br>`}`<br><br>5) Учениците по желба додаваат нови елементи, вметнуваат позадини (боја, слики или текстури) и стекнуваат рутина во пишувањето CSS правила преку комбинирање и експериментирање со ново изучените атрибути и нивните вредности. |
| • **Основни селектори во CSS (Елемент, идентификатор и класа)**<br><br>(Element. ID. Class.)<br><br>Број на часови: 2 (14, 15) | Во досегашните вежби и задачи кога сакавме да стилизираме некој елемент, до него пристапуваме на тој начин со тоа што се повикувавме со неговото име содржано во ознака. На пр. во параграфот ***<p>***, пристапуваме со тоа што ја пишуваме неговата буква ***"p { ... }"*** или ако сакаме да стилизираме наслов ***<h1>***, го земаме името на ознаката, т.е. ***"h1 { ... }"***.<br><br>Наставникот поставува прашања за размислување:<br><br>○ Дали овој начин е најфлексибилен и дали е доволен за комплетно да се стилизира една веб-страница?<br><br>○ Како да пристапиме или да го уредиме со посебни стилови **само** вториот или третиот параграф во страницата?<br><br>По дадените одговори и донесените заклучоци, наставникот им ги презентира двата нови начини за селектирање на елемент, односно како функционираат, кои се нивните карактеристики и нивната правилна синтакса во **html** и **css**.<br><br>○ Селектор по идентификатор – **(ID):** Се користи **само за еден, уникатен елемент** со одреден стил на уредување, односно дефинираниот стил го добива само тој елемент и никој друг. |

- Селектор по класа **(class):** Се користи за **стилизирање на повеќе елементи**, односно повеќе елементи можат да ги добијат стиловите дефинирани во таа класа.

| Карактеристика | селектор ID | селектор CLASS |
|---|---|---|
| Синтакса во HTML | **id** = "ime" | **class** = "ime" |
| Синтакса во CSS | **#**ime { ... } | **.**ime { ... } |
| Повторување во HTML | Само **еднаш** по документ | Повеќепати |
| Префикс во CSS (повикување) | Хаштаг/тараба **#** | Точка **.** |
| Употреба | За еден, **уникатен** елемент | За **повеќе** елементи |
| Пример употреба во HTML | **<h1** id="naslov"**></h1>** | **<p class="**tekst"**></p>** |
| Пример употреба во CSS | **#naslov** { color: red; } | **.tekst** { font-size: 18px; } |
| Примена на повеќе елементи | НЕ | ДА |

**Задача за учениците:**

1) Се креира нова **html** датотека и се зачувува по името **selectors.html**. Се копира содржината од фајлот ***backgrounds.html***.

2) Во **html** документот во ознаката **h1** се додава **ID** идентификатор под името **“naslov”**. Во делот **<style>** селекторот по “име” **h1** се заменува со името на ID селекторот **#naslov**.

```
<!-- Синтакса во HTML -->
   <h1 id="naslov">Ова е наслов напишан со фонт преземен од Google Fonts</h1>

 /* Синтакса во CSS */
   #naslov { font-weight: 100; font-style: italic; background-color: #abebc6; font-size: 56px; }
```

3) Направи го текстот во вториот параграф со црна позадина и бел текст!

Во конкретниот случај за вториот параграф ќе креираме **класа** под име **“*.dark-mode*”**во која ќе

се зададат CSS правила за црна/темна позадина и светла боја на текстот. Истата класа ќе може во иднина да се примени и на некој друг параграф што ќе биде додаден во документот.

```
<!-- Синтакса во HTML -->
    <p class="dark-mode">Lorem ipsum dolor sit amet,... </p>

  /* Синтакса во CSS */
    .dark-mode { background-color: black; color: orange; font-weight: 300; }
```

Учениците можат и по нивна желба да ја изберат бојата на текстот, сè додека е јасно видлива на црната позадина.

4) Во документот може да се забележи дека **<h1>** и **<p>** делат заеднички карактеристики, како што се фонтот на текстот (*Roboto*) и неговото порамнување (*средина*). Сето тоа можеме да го ставиме во класа, која ќе може да се примени на двата елементи. Во тој дел имињата **h1** и **p** се заменуваат со класа под името ***.roboto-center***. За да работи правилно името на класата се става и во соодветните ознаки во **html** структурата. Оваа класа секогаш можеме да ја искористиме во елементи каде што сакаме текстот да биде испишан во фонтот *Roboto* од *Google Fonts* и порамнет на средина.

```
<!-- Синтакса во HTML -->
    <h1 id="naslov" class="roboto-center">Ова е наслов напишан со фонт преземен од Google Fonts</h1>
    <p class="roboto-center">Lorem ipsum dolor sit amet,... </p>
    <p class="dark-mode roboto-center">Lorem ipsum dolor sit amet,... </p>

  /* Синтакса во CSS */
    .dark-mode { font-family: "Roboto", text-align: center; }
```

Наставникот го коментира крајниот изглед на елементите во структирата на **html** документот:

- Еден елемент истовремено може да содржи и **id** и **class** (**<h1>** ги содржи **#naslov** и **.roboto-center**)
- Еден елемент може да содржи повеќе класи (вториот параграф **<p>** ги содржи **.dark-mode** и **.roboto-center** ставени во наводници одделени само со празно место)

- **Маргини, рамки и внатрешно растојание**

  (Margin. Border. Padding)

  Број на часови: 4 (16 – 19)

Еден од основните концепти во веб-дизајнот е таканаречениот ***CSS Box-Model.*** Овој фундаментален концепт го објаснува начинот на кој се пресметува димензијата и просторот на секој HTML елемент. Секој елемент на веб-страницата се третира како правоаголна "**кутија**" (или **box**), а box-моделот дефинира како овие кутии се формирани и како се контролира просторот околу нив. Познавањето на box-моделот е од клучно значење за прецизно форматирање и распоредување (позиционирање) на елементите во страницата.

**Box**-моделот, односно елементите се состојат од четири основни компоненти:

1) **Содржина (Content)**

- Ова е внатрешниот дел од "кутијата" т.е она што го гледа корисникот, како што се текст, слики, видеа или други HTML елементи.
- Големина на оваа компонента се дефинира својствата: **width** и **height**.

2) **Внатрешна маргина (Padding)**

- Ова е просторот помеѓу содржината и работ на кутијата (border). Служи за да се направи содржината повидлива и да се избегне нејзино "лепење" до работ т.е. **рамката**.
- Може да се постави за сите страни (горе, долу, лево, десно) или поединечно.
- Ги прима следните својства: *padding, padding-top, padding-right, padding-bottom, padding-left*.

3) **Рамка (Border)**

- Ова е линијата што ја опкружува содржината и пополнувањето и служи за визуелно раздвојување на елементите.
- Може да се стилизира со различни бои, дебелина и стилови (**solid** - полна, **dashed** - испрекината, **dotted** - со точки, итн.).
- Ги прима следните својства: *border, border-width, border-style, border-color*.

4) **Маргина (Margin)**

- Ова е надворешниот простор околу работ на кутијата и служи за да се создаде растојание помеѓу еден елемент и другите елементи на страницата.

- Маргините не се дел од самата кутија, туку се простор надвор од неа.
- Ги прима следните својства: *margin, margin-top, margin-right, margin-bottom, margin-left*.

**Задача за учениците:**

1) Се креира нова ***html*** датотека и се зачувува по името ***box-model.html***. Се копира содржината од фајлот ***selectors.html***.

2) Со десен клик на веб-страницата, од менито се избира опцијата ***Inspekt*** за да се пристапи до делот ***DevTools*** (*Developer Tools, види слика на стр. 28*). До овој дел исто така, може директно да се пристапи преку пристискање на копчето ***F12*** од тастатура.

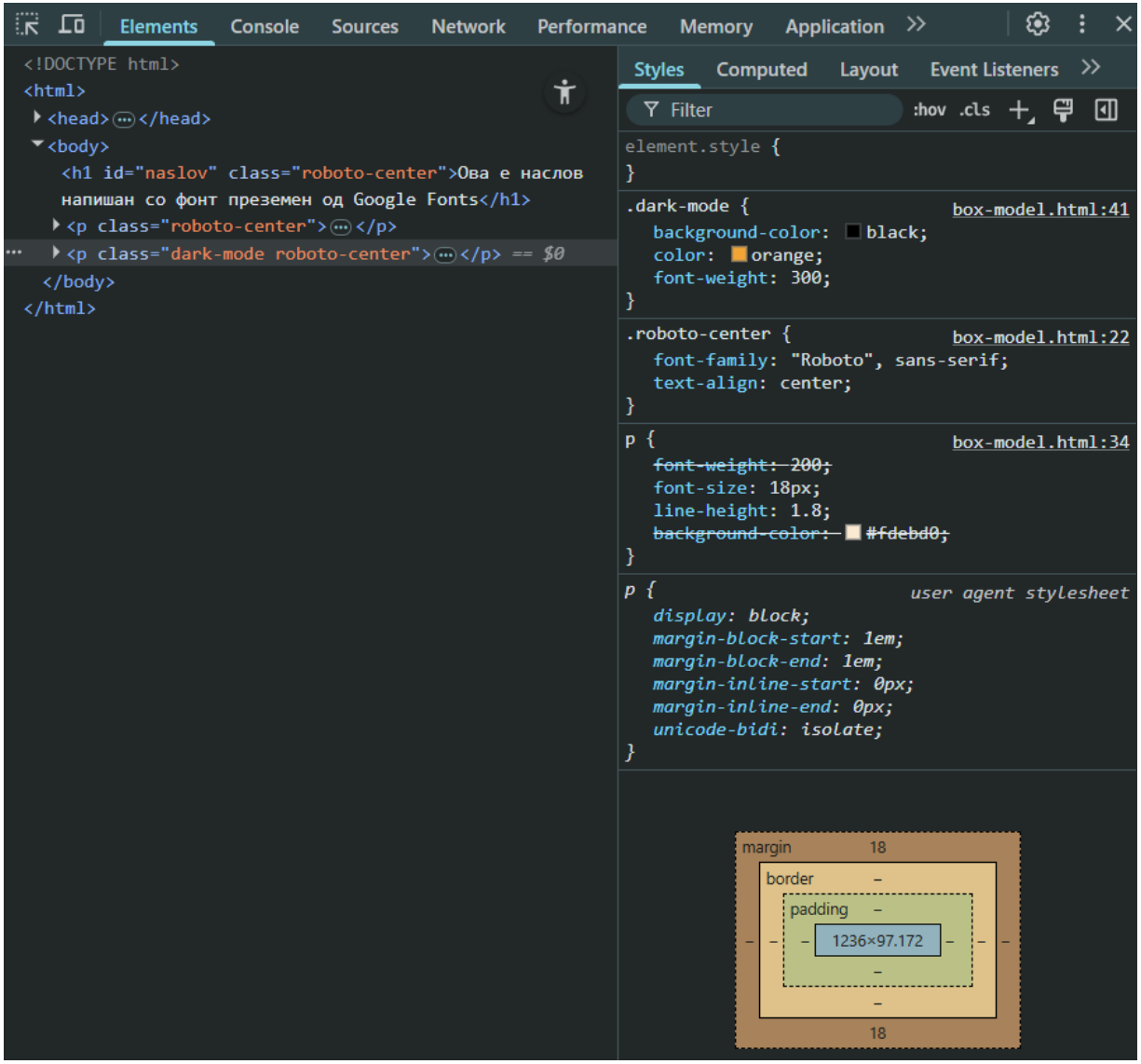
- ! **DevTools** е збир на алатки, вградени во веб-прелистувачот Google Chrome. Тие му овозможуваат на програмерот/веб-развивачот детално да ја прегледа, уредува и дебагира веб-страницата во реално време, директно од прелистувачот.

3) Се пристиска на икона  ***Select an element in the page to inspect it – Ctrl + Shift +C,*** а потоа со помош на глувчето се одбира елементот што сакаме да го разгледаме.

4) Од структурата на правоаголници, позиционирана во долниот десен агол од ***DevTools,*** се забележува дека двата елементи **<p>** делат исти својства, т.е. имаат иста вредност од 18px за маргините горе и долу. Компонентите: *padding, border* и **маргините лево и десно** (margin-left, margin-right) се 0px, односно стандардната "фабричка" вредност за параграф. Со помош на горенаведените компоненти или атрибути на box-моделот можеме да ги уредуваме овие вредности.

- Правоаголникот со боја (светло-сина), ја претставува содржината (***content***) на елементот (текст, слика...)
- Правоаголникот со боја (светло-зелена), ја претставува внатрешната маргина (***padding***)
- Правоаголникот со боја (светло-жолта), ја претставува рамката (***border***)
- Правоаголникот со боја (светло-кафеава), ја претставува надворешната маргина (***margin***)

5) Се разгледува и насловот **h1**, се забележуваат неговите карактеристики, односно горна и долна маргина од **37px**.




**Слика 1.** ***Chrome Developer Tools***

Се поставуваат прашања во однос на меѓусебното однесувањето на маргините помеѓу два соседни елементи:

- Насловот има долна маргина од 37px, а првиот параграф веднаш под насловот има горна маргина од 18px. Дали растојанието помеѓу двата елементи ќе биде 55px, односно колку што изнесува збирот од двете соседни маргини?
  ***Одговор:*** *Не, растојанието е 37px.*

- Двата параграфи имаат соседни маргини од 18px. Дали вкупното растојание помеѓу нив изгледа како 36px?
  ***Одговор:*** *Не, растојанието е 18px.*

Оваа појава се нарекува ***Margin Collapse*** и треба да се земе во предвид при дизајнирањето на распоредот на елементите. Таа претставува посебен механизам во CSS што го регулира начинот на кој маргините се комбинираат кај соседните елементи. Ова значи дека, ако имаме два елементи еден под друг, каде што првиот има **margin-bottom:** 20px, а вториот има **margin-top:** 30px, вкупниот простор помеѓу нив нема да биде **50px**, туку поголемата од двете вредности **(30px)**.

За полесно да се совлада материјалот за box-моделот ќе се користат вежби со ознаката ** ** во HTML. Ознаката ** ** (од "*division*" - *поделба*) генерички е **блок-елемент*** во HTML што се користи за групирање (поделба) на содржина. Сам по себе, ** ** нема никакво семантичко значење.

* **! Во HTML, елементите се делат на блок-елементи (block-level) и во линиски (inline) според нивното однесување во документот. Block-level ( , <p>, <h1> до <h6>, <ul>, <ol>, <li>, <section>, <article>, <header>, <footer>, <main> итн.) елементите секогаш започнуваат во нов ред и ја заземаат целосната ширина на елементот во кој се вгнездени. Inline (<a>, <img>, , , , <label>, <b>, <i> итн.) елементите не започнуваат во нов ред и заземаат простор онолку колку што им е потребно. Можат да се користат како вгнездени елементи внатре во block-level, но не можат да содржат блок-елемент.**

**Употреба:**

- **Групирање на содржина:** Најчесто се ко користи за логичко групирање на поврзани елементи.

- **Стилизирање со CSS:** Бидејќи ** ** нема вграден визуелен приказ, најголемата придобивка доаѓа се употребува во CSS. Со додавање на ***class*** или ***id*** атрибути на ** ** елементот, може да се стилизираат елементи (боја, фонт, распоред, големина, итн.), со што се овозможува создавање на сложени распореди и дизајни на веб-страници.

- **Распоред (Layout):** Иако во HTML5 има поспецифични ознаки (<header>, <nav>, <main>, <article>, <section>, <footer>), ** ** сè уште е незаменлив во многу аспекти на распоредот, особено во flex-box и grid-layout распоредите.

- **JavaScript манипулација:**  елементите често се користат за манипулација со ***JavaScript***, овозможувајќи динамични промени на содржината или стиловите на веб-страницата.

```
<!-- Синтакса во HTML -->
    <div class="card">
            <h2>Наслов на картичката</h2>
            <p>Ова е параграф со некоја содржина во картичката.</p>
            <button>Прочитај повеќе</button>
    </div>
```

Во овој пример, ** ** со класа "card" ги групира насловот, параграфот и копчето, овозможувајќи им да бидат стилизирани заедно како една визуелна "картичка" со помош на CSS.

**Задача за учениците:**

1) Во документот ***box-model.html*** креирај два елементи ** ** и додади класа ***.box1*** и ***.box2*** соодветно на двата елементи. Во секој од двата елементи додади содржина, односно произволен текст (пр. Квадрат бр. 1, Квадрат бр. 2, итн).

```
<!-- Синтакса во HTML -->
    <div class="box1">Квадрат бр. 1</div>
    <div class="box2">Квадрат бр. 2</div>
```

2) Се креираат CSS правила за класите **.box1** и **.box2**. Користејќи ги атрибутите **weight** и **height** се дефинира **ширина** и **висина** од **300px**, со цел од елементот ** ** да се креира квадрат (“кутија”). Се дефинира и боја на секоја од креираните квадрати со помош на **background-color**.

3) Во **.box1** класата дефинираме дополнителни CSS правила: додаваме маргини од по **50px** на сите четири страни од квадратот, рамка со дебелина од **5px**, **полна линија** и **црна** боја.

```
/* Синтакса во CSS */
    .box1 {
            width: 300px;
            height: 300px;
            background-color: orange;
            margin: 50px;
            border: 5px solid black;
           }

    .box2 {
            width: 300px;
            height: 300px;
            background-color: olivedrab;
           }
```

4) Квадратот со класа **.box1** добива маргини и црна рамка со полна линија од СИТЕ четири страни. Во ***DevTools*** со *Inspect Elements* го разгледуваме првиот квадрат: Се забележува дека ширината и висината на квадратот сега е 310x310px, поради дебелината на рамката од 5px соодветно од двете страни, т.е. лево+десно и горе+долу.

5) Се додава ново CSS правило за внатрешна маргина **(padding)** со вредност 50px.

```
/* Синтакса во CSS */
    .box1 {
            width: 300px;
            height: 300px;
            background-color: orange;
            margin: 50px;
            border: 5px solid black;
            padding: 50px;
           }
```

6) Веднаш се забележува дека текстот се оддалечува кон внатрешноста од рамката по **50px** од сите страни. Визуелно се гледа и дека квадратот дополнително ги зголемува неговите димензии. Во **DevTools** со *Inspect Elements*, се забележува дека новата големина на квадратот сега е 410x410px, поради додадената внатрешна маргина помеѓу содржината и рамката од сите страни.

ВАЖНО**!** Во **стандардниот** box-модел, ако на пр. се постави ***width:*** *200px,* на еден елемент, неговата вистинска ширина ќе биде поголема заради ***padding*** и ***border***. Пресметувањето на вкупната ширина **(width)** и висина **(height)** на елементот се пресметува на следниот начин:

- Вкупна ширина = **width + padding-left + padding-right + border-left + border-right**
- Вкупна висина = **height + padding-top + padding-bottom + border-top + border-bottom**

Токму поради ова воведен е и ***Алтернативен box-модел (Alternative Box Model),*** кој се активира со **box-sizing:** *border-box*. Во овој модел, атрибутите **width** и **height** автоматски се прилагодуваат вклучувајќи ги **padding** и **border**. Што значи, големината на содржината (content) автоматски се пресметува и прилагодува за да се вклопи во зададената ширина и висина на елементот. На пример: Ако поставиме **width:** 200px на еден елемент, со **box-sizing:** *border-box*, неговата крајна ширина ќе биде точно 200px.

7) Се додава ново CSS правило “**box-sizing:** *border-box”* во класата **.box1** и во **DevTools** со *Inspect Elements* се разгледуваат новите карактеристиките на првиот квадрат. Се забележува дека

квадратот ги задржал првичните поставки за висина и ширина (300x300px), и е позициониран **50px** лево во однос на **<body>** т.е неговиот "родителски" елемент во кој што е вгнезден и **50px** од горе во однос на вториот параграф.

```
/* Синтакса во CSS */
    .box1 {
            ...
            ...
            box-sizing: border-box;
          }
```

! И во овој модел постои ***Margin Collapse***, или појава каде што имаме "преклопување" на маргините. Иако вториот параграф има долна маргина од **18px**, се зема поголемата вредност од **50px** за растојание помеѓу двата елементи.

**Табеларен приказ за CSS синтакса на:** margin / padding

| Број на вредности | Пример | Значење (редослед) |
|---|---|---|
| 1 | **margin:** 10px; | Сите 4 страни (горе, десно, долу, лево) добиваат маргина од 10px |
| 2 | **margin:** 10px 20px; | 1. Горе и долу: 10px<br>2. Лево и десно: 20px |
| 3 | **margin:** 10px 20px 30px; | 1. Горе: 10px<br>2. Лево и десно: 20px<br>3. Долу: 30px |
| 4 | **margin:** 10px 20px 30px 40px; | 1. Горе: 10px<br>2. Десно: 20px<br>3. Долу: 30px<br>4. Лево: 40px |

Редоследот на страните кај атрибутот ***margin / padding*** со четири вредности е следен: Top → Right → Bottom → Left (горе, десно, долу, лево). Истата синтакса важи и за ***padding***.

Доколку сакаме да поставиме надворешна или внатрешна маргина специфично само на една

страна се користат следните атрибути:

- **За надворешна маргина:** *margin-top, margin-right, margin-bottom, margin-left*.
- **За внатрешна маргина:** *padding-top, padding-right, padding-bottom, padding-left*.

Пример: **margin-right:**10px, прави маргина од **10px** само на десната страна од елементот. Истото важи и за ***padding***.

**Табеларен приказ за CSS синтакса на:** border

| Својство | Пример | Опис |
|---|---|---|
| border | **border**: 1px solid black | **Скратена синтакса:** дебелина, стил (*solid, dotted, dashed, double*), боја – се применува на сите 4 страни |
| border-width | **border-width:** 2px 4px; | Ширина на рамка (исто како кај margin/padding),<br>Прима од 1 до 4 вредности |
| border-style | **border-style**: solid; | Стил на рамка: solid, dotted, dashed, double, none и др. |
| border-color | **border-color:** red green; | Боја на рамка (исто како кај margin/padding),<br>Прима од 1 до 4 вредности |
| border-top | **border-top:** 1px solid red; | Го дефинира само горниот раб |
| border-right | **border-right**: 1px dashed blue; | Го дефинира само десниот раб |
| border-bottom | **border-bottom:** 3px double green; | Го дефинира само долниот раб |
| border-left | **border-left:** 5px dotted purple; | Го дефинира само левиот раб |
| border-radius | **border-radius:** 10px; | Заоблување на аглите |

8) Во класата **.box1** додади го атрибутот **border-radius** со вредност **15px**. Се добива интересен изглед со заоблени агли. Обиди се да направиш круг од друг елемент.

9) Во декларацискиот блок од елементот **body,** додади ги следните CSS правила: **width:** 75%; (елементот ќе зазема **75% од екранот**) и **margin:** 0 auto; (css трик за центрирање на содржината

| | |
|---|---|
| | на средина од страницата).<br><br>10) Во вториот параграф односно класата **.dark-mode** збогати ја со css правило за додавање внатрешна маргина на елементот **padding** од **20px** и **border-radius** од **15px.**<br><br>11) Користејќи ги досега изучените атрибути и правила, стилизирај го вториот ** ** со класа **.box2**, како и останатите елементи од **html** документот. |
| • **Блок-елемент (block), линиски блок (inline-block) и флексибилен распоред (flex).**<br><br>(display. block. inline-block. flex)<br><br>Број на часови: 1 (20) | Овој час се започнува со предизвик за учениците: “Обиди се да ги подредиш двата квадрати хоризонтално еден до друг од документот **box-model.html.”**<br><br>1) Копирај ја содржината од **box-model.html** во нов документ именуван како **inline-block.html**<br><br>○ Како што споменавме претходно ознаката ** ** генерички е **блок-елемент** во HTML, односно блок-елементите секогаш започнуваат во нов ред, што значи кај него атрибутот **display** стандардно ја содржи вредноста **block**. За да го постигнеме ефектот што се бара во задачата потребно е да се зададе css правило во класите ***.box1*** и ***.box2*** за менување на вредноста на атрибутот **display** од *block* во *inline-block*.<br><br>*/* Синтакса во CSS */*<br>**.box1 {**<br>**…**<br>**display:** *inline-block;*<br>**}**<br><br>**.box2 {**<br>**…**<br>**display:** *inline-block;*<br>**}**<br><br>ВАЖНО**!** Кај елементи со **display:** inline-block не се јавува *Margin Collapse*. Оваа појава се случува само меѓу елементи со **display:** block и важи само за **вертикални** маргини (top и bottom), но не и за **хоризонтални** (left и right). Елементите со **display:** inline-block се **неподложни** на *Margin Collapse* и се однесуваат **како текст**, поставени едни до други хоризонтално и секој си ги задржува своите маргини. |

2) Откако ќе ги внесеш css правилата во **DevTools** со *Inspect Elements* провери ги новите карактеристики (горните маргини) на квадратите и соседниот параграф од горна страна.

3) Вториот начин е *"old-school",* односно користење на атрибутот **float.** Двата ** ** елементи кои ги содржат класите **.box1** и **.box2** се групираат односно, ќе бидат вгнездени во еден ** ** со класа .**container.** Едниот квадрат или **.box1** добива вредност **float:** left, а другиот **float:** right; Класата .container добива внатрешна маргина **padding** лево и десно од 200px. На овој начин двата квадрати не мора да бидат *inline-block* и тој код наместо да го бришеме, ќе го ставиме во коментар.

**Коментар во CSS се прави на следниот начин: ** Ова е коментар ** или се селектира кодот кој сакаме да го ставиме во форма на коментар и се притиска на комбинацијата од тастери Ctrl + Shift + /.**

```
<!-- Синтакса во HTML -->
    <div class="container">
            <div class="box1">Квадрат бр. 1</div>
            <div class="box2">Квадрат бр. 2</div>
    </div>

/* Синтакса во CSS */
    .container {
                    padding: 0 200px;
                }
    .box1 {
            ...
            /* display: inline-block; */
            float: left;
            }
    .box2 {
            ...
            /* display: inline-block; */
            float: right;
            }
```

| | |
|---|---|
| | 4) Третиот начин за подредување на блок-елементи во хоризонтала е следен: Во класата **.container** се задава css правило **display:** flex. **Flex** е CSS својство/вредност што се користи за флексибилно распоредување на елементи во ред или колона. CSS правилата со **float** ги ставаме во коментар.<br><br>Атрибутот **display** кога има вредност **flex**, овозможува:<br><br>○ Лесно порамнување на елементите по хоризонтала или вертикала;<br>○ Распределување на простор меѓу елементи;<br>○ Прилагодување на димензии според достапниот простор;<br>○ Респонзивно распоредување на елементите без користење на float или position.<br><br>*/* Синтакса во CSS */*<br>**.container {**<br>**display:**flex*;*<br>**}**<br>**.box1 {**<br>**…**<br>**/* display: inline-block; */**<br>**/* float:** left; */<br>**}**<br>**.box2 {**<br>**…**<br>**/* display: inline-block; */**<br>**/* float:** right; */<br>**}** |
| • **Флексибилен распоред (Flexbox)**<br><br>(Flex. распоред. редица. колона. x-оска. y-оска) | Како што видовме на претходниот час, **Flexbox** (или **Flexible Box Layout**) е CSS-механизам за распоредување на елементите во редица или колона, со флексибилност при големината и просторот помеѓу нив. Главната предност на **Flexbox** е што овозможува лесно порамнување и распоредување на елементи во однос на нивниот “родител” (***parent element***). |

| | |
|---|---|
| Број на часови: 3 (21 – 23) | **Задача за учениците:**<br><br>1) Креирај нова **html** датотека и именувај ја како **flexbox.html**. Направи автоматско дополнување со основната **html** структура. (**html** + press **tab**). Опционално, вметни го и линкот од *google fonts* доколку сакаш да го користиш фонтот "*Roboto*" или избери си некој друг фонт.<br><br>Оваа задача ќе ја искористиме за да се запознаеме со третиот начин на вчитување на css датотека, односно **external** css (начинот кој го користат професионалниците за веб-развој и дизајн):<br><br>2) Наместо како досега, каде што пишувавме CSS правила во ознаката **<style></style>**, сега ќе користиме друг посебен документ кој ќе се користи за таа намена. Што значи, креираме нова датотека и ја именуваме **style.css**. Во делот **<head>** под линкот за вчитување фонт се внесува ознаката **<link>** **(link +** press **tab).** Се добива следниот формат: **<link rel="stylesheet" type="text/css" href="">.** Во атрибутот **href** во делот со наводници ја внесуваме адресата каде што се наоѓа **css датотеката**. Бидејќи, **html** и **css** датотеките споделуваат заедничка локација, го впишуваме само името на документот, т.е. **href= "style.css"**.<br><br>**!** Доколку фајлот се наоѓа на друга адреса, на пример, во папка именувана како **"css"**, го впишуваме следното: **href="css/style.css".**<br><br>3) Добра пракса е да се провери дали **css** документот е правилно поврзан со **html**. Затоа, во **style.css** задаваме css правило за менување на позадината на **<body>** со помош на **background-color**.<br><br>*/* Синтакса во CSS */*<br>**.body { background-color: bisque;** *}/* нијанса на светло-кафеава (беж) боја */*<br><br>4) За да видиме како функционира распоредувањето на елементите со помош на **Flexbox** креираме 3 квадрати со исти или слични карактеристики за: димензии, боја, рамка и текст. Поставуваме "родител" елемент ** ** со класа **.container**, а останатите ** ** елементи од кои ќе бидат формираат квадратите добиваат по две класи: една заедничка **.box** и по една за секоја |

одделно: ***.item-1***, ***.item-2*** и ***.item-3***.

```
<!-- Синтакса во HTML -->
<body>
    <div class="container">
            <div class="box item-1">1</div>
            <div class="box item-2">2</div>
            <div class="box item-3">3</div>
    </div>
</body>
```

5) Во датотеката **style.css** ги поставуваме **css** правилата за стилизирање на елементите:

```
/* Синтакса во CSS */

*,  *::before, *::after {
box-sizing: border-box;
}

body {
    background-color: bisque;
    font-family: "Roboto", sans-serif;
    font-size: 5em;
    color: #f1f5f9;
    }

.container {
    margin: 0 auto;
    width: 80%;
    height: 1200px;
    border: 10px solid #f5b027;
    border-radius: 10px;
    display: flex;
    }
```

```
.box {
    width: 300px;
    height: 300px;
    background-color: #fb7185;
    border: 10px solid #e11d48;
    border-radius: 10px;
    }
```

6) **Flexbox** распоредот за елементи ги има следните атрибути:

- **flex-direction:** row / column / row-reverse / column-reverse (распоредување по x-оска/y-оска или по хоризонтала/вертикала)
- **justify-content**: flex-start / center / flex-end / space-between / space-around / space-evenly (порамнување на елементите и подесување на растојанието помеѓу нив)
- **align-items:** flex-start / center / flex-end (порамнување по y-оска или по вертикала)

Наставникот ги презентира крајните исходи од фукциите на секоја од наведените атрибути и нивните соодветни вредности. Потоа, учениците во декларацискиот блок од класата **.container** ги внесуваат наведените атрибути, ги комбинираат и преку менување на нивните вредности ги согледуваат крајните резултати од распоредот на елементите.

7) Дел од останатите атрибути од **Flexbox layout**:

- **Flex-wrap:** no-wrap / wrap (прилагодување на поставеноста на елементот во зависност од големината на прозорецот – респонзивност на елементите)
- **Gap:** 1em / 2em / 10px / 99px / итн (дефинирање на фиксен простор или растојание помеѓу елементите)
- **Flex-grow**: 1 (се определува кој елемент да “доминира”, од аспект на неговата големина во редоследот)

```
/* Синтакса во CSS */
    .
    .
    .
.container {
    margin: 0 auto;
    width: 80%;
    height: 1200px;
    border: 10px solid #f5b027;
    border-radius: 10px;
    display: flex;
    /*flex-direction: column; */
    /*justify-content: space-evenly; */
    /*align-items: center; */
    flex-wrap: wrap;
    gap: 1em;
    }

.item-2 {
    flex-grow: 1;
    }
```

8) **Предизвик за учениците**: “Обиди се да го позиционираш текстот во средината на квадратите, со помош на **Flexbox**”!

```
.box {
    ...
    display: flex;
    justify-content: center;
    align-items: center;
}
```

| | |
|---|---|
| • **Мини-проект: Изработка на веб-страница според претходно зададени параметри.**<br><br>(Мини-проект)<br><br>Број на часови: 1 (24) | Активностите предвидени во часовите за изработка на мини-проект (според зададени параметри) ќе послужат за утврдување на стекнатите знаења, односно стекнување рутина во пишувањето на CSS правила и развивање вештини во комбинирање на досега изучените елементи, атрибути и вредности во постапката за креирањето на една веб-страница. Пропратно ќе се изучат и нови дополнителни работи кои ќе допринесат за подобрување на севкупниот визуелен изглед на веб-страницата, како и воведување на поголема интерактивност и повратна информација помеѓу веб-страницата и корисникот. Исто така, целта на овој мини-проект е да се зголеми интересот кај учениците, да стекнат вештини за самостојно учење и критичко размислување и да послужи како дополнителна инспирација за наредниот голем проект кој ќе биде на тема по сопствен избор. Наставникот задава упатства за работа, развива дискусија, ги мотивира учениците да истражуваат и самостојно да се надоградуваат. Воедно, ја следи работата и стои на располагање за одредени нејаснотии кои евентуално би се појавиле за време на активностите.<br><br>**Задача за учениците:**<br><br>1) Креирај нова папка и именувај ја "**мини-проект**". Креирај нова **html** датотека под името **home.html**. Направи автоматско дополнување со основната HTML структура (**html** + press **tab**). Во **<title></title>** внеси го текстот "*Mini-project*". Во делот **<head>** вчитај линк од ***Google Fonts*** за фонтовите *Rotobo* и *Tektur*. Креирај нова датотека **style.css** и вчитај ја како ***external css*** во делот **<head>** со помош на ознаката **<rel>**.<br><br>2) Се започнува со дефинирање на структурата во **html**. Се додаваат следните ознаки:<br><br>○ **<header></header>** со **id="header".** Заглавието (*header*) ќе содржи еден наслов од ниво **<h1></h1>** со текст "ДОБРЕДОЈДОВТЕ НА МОЈОТ ВЕБ-САЈТ";<br><br>○ **<nav></nav>** која ќе содржи листа од која ќе се изработи мени за навигација низ веб-страницата;<br><br>○ ** ** со класа **class="container"**. Овој дел ќе биде резервиран за содржината во веб-страницата (слики, текстови, икони, копчиња итн.); |

- **<footer></footer>** HTML елемент за дефинирање долен дел на страницата (подножје). Обично содржи информации како авторски права (*copyright*), контакт информации и сл. Во нашиот случај ќе содржи текст вгнезден во параграф: ** ©2025, Hackerman - The most powerful hacker of all time. All rights reserved. **

```
<!-- Синтакса во HTML -->

<body>
    <header id="header">
            <h1>ДОБРЕДОЈДОВТЕ НА МОЈОТ ВЕБ-САЈТ</h1>
    </header>
    <nav>
    </nav>
    <div class="container">
    </div>
    <footer>
            <p>©2025, Hackerman - The most powerful hacker of all time. All rights reserved.</p>
    </footer>
</body>
```

3) Во документот **style.css** на почетокот со помош на **универзалниот селектор ***, се задава CSS код за поставување на алтернативниот **box-модел** (**box-sizing:** *border-box*) за сите елементи кои ќе ги содржи веб-страницата.

```
/* Синтакса во CSS */

    *,
    *::before,
    *::after {
            box-sizing: border-box;
            }
```

4) На https://www.transparenttextures.com/ Во делот за пребарување внеси ги зборовите “Brick Wall” и преземи ја текстурата под името **Brick Wall (Dark)** – Made by Benjamin Ward. Текстурата именувана како ***brick-wall-dark.png*** зачувај ја во нова папка под името **img**.

5) Стилизирај го **<body></body>** со следните **CSS** правила:

```
/* Синтакса во CSS */

    body {
            background-image: url("img/brick-wall-dark.png");
            background-color: white;
            font-family: "Tektur", sans-serif;
            width: 80%; /* ќе зазема ширина од 80% од видливиот дел на прозорецот од прелистувачот */
            margin: 34px auto;
            }
```

6) Се стилизира **<header></header>** преку селекторот по идентификатор **id=“header”**.

```
/* Синтакса во CSS */

    #header {
            display: flex;
            justify-content: center;
            align-items: center;
            height: 20vh; /* висината на елементот ќе биде 20% од висината на видливиот дел од прозорецот на
                             прелистувачот (viewport). */
            width: 80vw; /* ширината на елементот ќе биде 80% од ширината на видливиот дел од прозорецот на
                            прелистувачот (viewport). */
            border: 4px solid black;
            border-radius: 10px;
            }
```

7) Се стилизира **<h1></h1>** со следните карактеристики:

```
/* Синтакса во CSS */

h1 {
        font-size: 5em; /* Големината на текстот во насловот ќе изнесува 80px */
        color: black;
        }
```

**!** **5em** означува дека големина на текстот ќе биде 5 пати поголема во однос на големината на текстот на "родителскиот" елемент (*parent element*). Во случајот, родителскиот елемент од кој што се наследува големината, т.е. **<body>**, стандардно има **font-size: 16px**, што значи **5em = (5 × 16px = 80px).**

8) Подножјето (*Footer*) се стилизира со следните CSS правила:

```
/* Синтакса во CSS */

footer {
        position: fixed; /* CSS правило за поставување фиксна положба на дното од страницата*/
        left: 0;
        bottom: 0;
        width: 100%;
        background-color: black;
        color: white;
        text-align: center;
        font-size: 1.3em;
        }
```

- **Мини-проект: Изработка на мени за навигација**

  (Навигација. Листа. Мени.)

  Број на часови: 3 (25, 26, 27)

**Навигациското мени** е дел од веб-страницата кој им овозможува на корисниците да се "движат", односно да навигираат низ различни страници, секции или функции во веб-страницата. Навигацијата е стандарден дел, што го очекуваат сите корисници кога ќе ја посетат веб-страницата и помага при пронаоѓање на потребните содржини. Исто така, преку навигацијата се прикажува хиерархијата и организација на веб-страницата.

**Задача за учениците:**

1) За изработка на навигациското мени ќе користиме неподредена листа (*unorder list*) со помош на ознаката **<ul>**. Листата ќе содржи три артикли, т.е. **<li>.** Секој од нив ќе содржи ознака за линк (*anchor tag*) **<a>** со текстуална содржина (***ДОМА, ЗА МЕНЕ*** и ***ЛОКАЦИЈА***) и ќе води до соодветната секција од веб-страницата (***href*="#"**)

```
<!-- Синтакса во HTML -->

<nav>
        <ul>
            <li><a href="#">ДОМА</a></li>
            <li><a href="#">ЗА МЕНЕ</a></li>
            <li><a href="#">ЛОКАЦИЈА</a></li>
        </ul>
</nav>
```

2) Стандардно, ознаката **<ul>** има вредност **display**: block; т.е зафаќа цела хоризонтална линија или ред, а артиклите кои се содржани во неа се подредени вертикално. Горенаведената **html** структура ќе го даде следниот резултат:

- ДОМА
- ЗА МЕНЕ
- ЛОКАЦИЈА

Она што сакаме да го постигнеме е следното:

- Да се порамнат артиклите на средина во страницата;
- Да се подредат артиклите во хоризонтала;
- Да се тргнат “точките” кои се наоѓаат лево од текстот;
- Да се избрише долната линија под текстот;
- Да се направи простор/растојание помеѓу артиклите.

3) Ознаката **<nav>** добива нови вредности: **"display:** flex;**"** и **"justify-content:** center;**"**, за порамнување на целата листа на средина во страницата. Со цел подредување на артиклите во хоризонтала, ознаката **<ul>** добива вредност **"display:** flex;**"** а за бришење на точките од лева страна се користи **"list-style-type:** none; **"**. Долната линија под текстот се брише со **"text-decoration:** none; **"** преку пристапување до ознаката **<a>.** Растојанието помеѓу артиклите се дефинира со **"padding:** 0 60px; **"** во **<li>**.

```
/* Синтакса во CSS */
/* Menu */
nav {
    display: flex;
    justify-content: center;
    margin: 90px 0;
    }

ul {
    display: flex;
    font-size: 2.5em;
    font-weight: 400; /* Regular */
    color: black;
    list-style-type: none;
    margin: 0;
    padding: 0;
    }

li {
    padding: 0 60px; /* Внатрешни маргини: горе и долу 0px, лево и десно 60px за секој артикл */
    }

li > a {
    text-decoration: none;
    color: black;
}
```

4) Се прават уште две копии од датотеката **home.html**. Новите копии соодветно се преименуваат во **hackerman.html** и **location.html**. Се додава содржина на секоја од страниците во ознаката ** ** со класа **.container**:

- **home.html**

  - Се додава ознака **<h3>**Стискај ентер!**</h3>** и класа **. white-text-black-shadow**;
  - Се додава ознака **<i>** со соодветна класа за двојна стрелка надолу која ќе биде вчитана од надворешен извор: https://fontawesome.com/v4/; **class="fa fa-angle-double-down";** ВАЖНО**!** За да може да функционира вчитувањето на икони/симболи од специфичен надворешен извор со однапред дефинирани класи, во **<head>** задолжително мора да се стави следниот линк: <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
  - Се додава ознака за копче: **<button** type="button">ENTER**</button>.**

```
<!-- Синтакса во HTML -->
<div class="container">
    <h3 class="white-text-black-shadow">Стискај ентер!</h3>
    <div>
            <i class="fa fa-angle-double-down fa-2x"></i>
    </div>
    <button type="button">ENTER</button>
</div>
```

Се креира класа **.arrow,** се додава во ознаката **<i>** и се стилизира во CSS:

```
/* Синтакса во CSS */
/* Arrow */

.arrow {
    font-weight: bold;
    color: white;
    }
```

- **hackerman.html**

  - Се додава ознака **<img>** и се вчитува слика или **.gif** од популарниот **"meme"**, познат на Интернет под називот **"Hackerman - The most powerful hacker of all time"** (search on google 😉).

```
<!-- Синтакса во HTML -->
<div class="container">
    <img src="img/hackerman.gif">
</div>
```

- **location.html**

  - Се додава ознака **<h3>** localhost: 127.0.0.1**</h3>**
  - Се додава ознака **<i>** со соодветна класа **"fa fa-laptop fa-3x"** за икона од лаптот, вчитана од **https://fontawesome.com/v4/icons/**;

```
<!-- Синтакса во HTML -->

<div class="container">
    <h3>localhost: 127.0.0.1</h3>
    <i class="fa fa-laptop fa-3x"></i>
</div>
```

5) Се врши поврзување на линковите (врските) од менито со соодветната страница:

- **ДОМА** → **home.html**;
- **ЗА МЕНЕ** → **hackerman.html**;
- **ЛОКАЦИЈА** → **location.html**.

Во ознаката **<a>** во атрибутот **href="#"**, на местото од "хаштаг-от", помеѓу наводниците се става целосното име на страницата каде што сакаме да води линкот.

```
<!-- Синтакса во HTML -->

<nav>
        <ul>
            <li><a href="home.html">ДОМА</a></li>
            <li><a href="hackerman.html">ЗА МЕНЕ</a></li>
            <li><a href="location.html">ЛОКАЦИЈА</a></li>
        </ul>
</nav>
```

6) Се додаваат CSS правила за: **.container**, **h3,** .**white-text-black-shadow** и **button**.

```
/* Синтакса во CSS */
/* Content */

.container {
    display: flex;
    flex-direction: column;
    justify-content: flex-start;
    align-items: center;
    gap: 1.3em;
    width: 100%;
    height: 55vh;
    font-size: 4em;
    }

h3 {
    margin-top: 20px;
    margin-bottom: 0;
    }

.white-text-black-shadow {
    color: white;
```

```
    text-shadow: 5px 4px black; /* додавање ефект на сенка 5px по x-оска и 4px по y-оска */
    }

/* Button */

button {
    font-family: "Tektur", sans-serif;
    border: none;
    color: white;
    padding: 30px 64px;
    text-align: center;
    text-decoration: none;
    font-size: 26px;
    margin: 4px 2px;
    cursor: pointer;
    background-color: #04AA6D;
    letter-spacing: 3px;
    border-radius: 8px;
    }

/* Hackerman Image */

img {
    border-radius: 10px;
}
```

6) Од практична гледна точка, препорачливо е да се има некаков индикатор во менито за навигација кој ќе покажува која страница е активна во моментот. За таа цел во ознаката **<a>** додаваме класа **.active-menu** која ќе биде стилизирана со следните CSS правила:

```
/* Синтакса во CSS */

.active-menu {
    color: #04AA6D;
```

```
    font-weight: bold; /* 700 */
    text-shadow: 1px 1px black;
}
```

Ова значи дека линкот кој е активен ќе добие боја која одговара на следниот хексадецимален код **#04AA6D (нијанса на зелена боја)**, задебелен текст на фонтот и ефект на сенка во црна боја.

ВАЖНО! **Оваа класа се додава само во документот кој го содржи линкот кој води до него. На пример: Во документот HOME.HTML се додава класата .active-menu во линкот кој го содржи текстот ДОМА.**

```
<!-- Синтакса во HTML -->

<!-- home.html -->

<nav>
    <ul>
            <li><a class="active-menu" href="home.html">ДОМА</a></li>
            <li><a href="hackerman.html">ЗА МЕНЕ</a></li>
            <li><a href="location.html">ЛОКАЦИЈА</a></li>
    </ul>
</nav>

<!-- hackerman.html -->

<nav>
    <ul>
            <li><a href="home.html">ДОМА</a></li>
            <li><a class="active-menu" href="hackerman.html">ЗА МЕНЕ</a></li>
            <li><a href="location.html">ЛОКАЦИЈА</a></li>
    </ul>
</nav>
```

```
<!-- location.html -->

<nav>
    <ul>
            <li><a href="home.html">ДОМА</a></li>
            <li><a href="hackerman.html">ЗА МЕНЕ</a></li>
            <li><a class="active-menu" href="location.html">ЛОКАЦИЈА</a></li>
    </ul>
</nav>
```

- **Мини-проект: Користење на псевдо-селектори**

  (Псевдо-селектор. link. visited. hover. active.)

  Број на часови: 1 (28)

**Псевдо-селектори** (pseudo-selectors) се специјални селектори во CSS кои овозможуваат стилизирање на елементите во одредена **состојба** или **позиција**, без потреба од додавање класи или ID. Најчеста употреба на псевдо-селекторите е за стилизирање на линкови (**<a>** ознаката), каде што имаат клучна улога во **креирањето на интерактивно и динамично** корисничко искуство. На пример, која состојба или стил да се примени врз елементот кога корисникот го поставува покажувачот од глушецот врз него.

Краток опис на најважните псевдо-селектори за стилизирање на линкови:

- **:link** - селектира **непосетен линк**.
  - Овој селектор се применува на сите линкови на страницата кои корисникот сè уште ги нема кликнато. Стандардните стилови за линкови (на пр., сина боја и подвлечен текст) се дефинирани со овој селектор. Пример: **`a:link { color: blue; }`**
- **:visited** - селектира **веќе посетен линк**.
  - Откако корисникот ќе кликне на линкот и ќе ја посети страницата, прелистувачот ја менува неговата состојба во "visited". Овој селектор овозможува да се даде визуелна повратна информација на корисникот дека веќе бил на таа страница (на пр., со менување на бојата во виолетова). Пример: **`a:visited { color: purple; }`**
- **:hover** - селектира елементот **врз кој се наоѓа покажувачот на глувчето**.

- Се применува на линкот или кој било друг елемент кога корисникот го движи покажувачот од глувчето над него. Ова е клучно за интерактивност и визуелно истакнување на елементот. Честа практика е да се менува бојата, да се додаде или отстрани подвлекувањето или да се додаде сенка на елементот и слично. Пример: **a:hover { color: red; text-decoration: none; }**

- **:active** - селектира елемент **кој е моментално кликнат (активен)**.

  - Оваа состојба се активира кога корисникот ќе го притисне копчето на глувчето врз линкот, но сè уште не го отпуштил. Таа трае многу кратко, но е корисна за да се даде моментална визуелна повратна информација, на пример, со промена на бојата или додавање сенка за да се симулира „притиснато“ копче. Пример: **a:active { color: orange; }**

ВАЖНО! За правилно функционирање, препорачаниот редослед за овие псевдо-селектори е**: LVHA (Link, Visited, Hover, Active)**. Ако не се запази овој редослед, некои стилови може да не се применат како што се очекува.

**Задача за учениците:**

1) Во датотеката **home.html** ќе се применат псевдо-селектори на менито за навигација (**:hover** и **:active**) и на копчето "ENTER" (**:hover**) на дното од страницата.

2) **Текст од линкот во менито за навигација**: Во ситуација кога покажувачот на глувчето се наоѓа над текстот од линкот, да се прикажат горна и долна линија со црна боја (***black***) и дебелина од **3px**, а кога ќе се притисне на линкот да се смени бојата на текстот во **#04aa6d** (нијанса на зелена боја) и да се прикаже сенка поставена по **x** и **y** оска во вредност од **1px** со црна боја (***black***).

```
/* Синтакса во CSS */

a:hover {
    border-bottom: 3px solid black;
    border-top: 3px solid black;
    }
```

| | |
|---|---|
| | ```<br>a:active {<br>  color: #04AA6D;<br>  text-shadow: 1px 1px black;<br>}<br>```<br><br>3) **Копче "ENTER"**: Кога покажувачот на глувчето се наоѓа над копчето да се добие ефект на сенка, но од сите 4 страни:<br><br>```<br>/* Синтакса во CSS */<br><br>button:hover {<br>  box-shadow: 0 0 5px black;<br>  }<br>```<br><br>Учениците истражуваат на темата и по сопствен избор пишуваат нови css правила за избраните елементи, со цел додавање дополнителна интерактивност и динамика на веб-страницата. |
| • **Мини-проект: Анимации**<br><br>(animation. @keyframes.)<br><br>Број на часови: 1 (29) | Анимациите во CSS им овозможуваат на елементите во веб-страницата **да се движат, да ги менуваат нивниот облик, бојата или позицијата** во единица време, без потреба од користење **"JavaScript"**. Со нив, содржината станува **повизуелна, поинтерактивна и поинтересна** за корисникот.<br><br>Анимациите придонесуваат за **динамичен изглед**, се комбинираат со **псевдо-селектори**, како на пример со **:hover**, или **:active** при клик со покажувачот од глушецот. Се користат и за подобрување на **употребливоста и визуелната естетика** на веб-страницата. CSS анимациите се дефинираат со правила како **@keyframes** и атрибути/својства како *animation-name*, *animation-duration*, *animation-iteration-count*, *animation-delay*, *animation-direction*, итн. |

**Задача за учениците:**

Во веб-страницата од Мини-проектот се изработат следните анимации:

1) Сенката во текстот **<h3>** во почетната страница **home.html** ќе се придвижува нагоре (или надолу, налево или надесно) кога покажувачот од глушецот ќе биде позициониран над него.

- Се креира класа **.animation-shadow** во **<h3>** за пристапува до текстот и креирање на анимација. Се комбинира со псевдо-селекторот **:hover** и се дефинира името на анимацијата, нејзините основни вредности за името, времетраењето и однесувањето на анимацијата и поставките во **@keyframes.**

```
/* Синтакса во CSS */

/* Text Animation */

.animation-shadow:hover {
    animation: premesti-senka 1.4s infinite alternate;
    cursor: pointer;
    }

@keyframes premesti-senka {
    from { color: white; }
    to { color: white;  text-shadow: 6px -80px black; }
}
```

2) Стрелката која покажува кон копчето "ENTER" ќе добие периоднични придвижувања во негова насока во кратки интервали.

- Се креира класа **.arrow-bounce** и се додава во соодветната ознака **<i>**. Се дефинираат основните вредности во класата, име, времетраење и однесување на анимацијата, како и потребните поставки во **@keyframes**.

```
/* Синтакса во CSS */
/* Arrow Animation */

.arrow-bounce {
    animation: bounce 2s infinite;
    }
@keyframes bounce {
    0%, 20%, 50%, 80%, 100% {
     transform: translateY(0);
    }
    40% {
    transform: translateY(-30px);
    }
    60% {
     transform: translateY(-15px);
    }
}
```

3) Во страницата **location.html** во текстот од **<h3>** на крајот од текстот ќе се додаде **долна црта** ( _ ) **што трепка при пишување на компјутер** или т.н. "*блок-курсор*" или "*underscore cursor*", односно симулација на текстуален курсор во терминал или командна линија (CLI) .

- Долната црта која ќе се користи за симулација на блок-курсор се става во ознака **** на крајот од текстот во **<h3>** и се креира класа **.underscore-cursor.**

```
<!-- location.html -->

<h3>localhost: 127.0.0.1<span class="underscore-cursor">_</span></h3>
```

- Во декларацискиот блок од соодветната класа се дефинира името, вредноста за времетраење на анимацијата како и својства на однесување на анимацијата и потребните поставки во **@keyframes**.

```
/* Синтакса во CSS */
/* Customize Underscore Cursor */

.underscore-cursor {
    animation: blinking 0.5s infinite alternate;
    }

@keyframes blinking {
    from { color: black; }
    to { color: transparent; }
}
```

Учениците истражуваат на темата и по сопствен избор пишуваат нови css правила за поставување анимација на избраните елементи.

**Тема 3. ФИНАЛЕН ПРОЕКТ – Изработка на Веб-страница (7 часа)**

***Знаења/вештини:***

- Дизајнира сопствена веб-страница со употреба на HTML и CSS;
- Применува соодветни фонт-стилови, бои и распоред на содржини за визуелна прегледност;
- Додава слики, линкови и стилови за подобрување на изгледот на веб-страницата;
- Структуира и стилизира содржини користејќи класи и идентификатори;
- Знае за **box model** и како се користат margin, padding, border, width и height;
- Ги знае основите на **layout** со display: block, inline, inline-block и flex;
- Користи **flexbox** за прецизен и адаптивен распоред на елементи;
- Експериментира со **responsive** дизајн преку vh, em, rem и слично;
- Користење и креирање на основни **анимации** во CSS.
- Тестира и коригира грешки во HTML и CSS кодот;
- Презентира свој проект пред другите, објаснувајќи го процесот на работа.

***Ставови/вредности:***

- Ја почитува различноста во идеите и креативниот израз на секој ученик при изработката на веб-страницата.
- Смета дека финалниот проект е можност за применување на практични вештини и поттикнување на самодовербата.
- Ја прифаќа важноста на самостојната и тимската работа при реализирање на проектот.
- Подготвен/а е да презема активности кои вклучуваат планирање, изработка, тестирање и презентација на веб-страница.
- Има позитивен став кон учење преку проекти, каде што учениците можат да ја демонстрираат својата креативност и техничко знаење.
- Ја осудува неетичката практика на копирање туѓа работа без дозвола и ја поддржува оригиналноста.
- Верува дека секој ученик може да создаде корисна и визуелно привлечна веб-страница со вложен труд.
- Поддржува со користење на научените HTML и CSS концепти за реална примена во дигиталниот свет.
- Разбира дека финалниот проект не е само оценка, туку и демонстрација на личен развој и стекнати дигитални вештини.
- Се залага за отворена и мотивирачка училишна средина каде што се вреднува практичната примена на знаењето.

| *Содржини (и поими) и број на часови* | *Примери на активности:* |
|---|---|
| • **Изработка на веб-страница на тема по сопствен избор.**<br>(Веб-страница)<br><br>Број на часови: 6 (30 - 35) | **Тема:** Изработка на веб-страница со примена на досега изучените концепти од HTML и CSS**.**<br>**Цел:** Учениците (во парови или тимски) да изработат сопствена веб-страница како финален продукт, применувајќи ги стекнатите знаења и вештини.<br><br>**1) Претставување на проектната задача**<br><br>Наставникот ги информира учениците дека започнуваат со работа на финалниот проект и објаснува што треба да содржи веб-страницата (минимални барања и слобода за креативност). Презентира пример веб-страница за инспирација и ги коментира нејзините делови (структура, стил, организација итн). Ги споделува и појаснува критериумите за оценување (структура, изглед, читливост, креативност, коректност на кодот) и истовремено ги охрабрува учениците да поставуваат прашања за барањата и очекувањата.<br><br>**2) Оформување на работни тимови и поддршка при избирање на тема**<br><br>Наставникот го организира работниот распоред (во парови или мали групи) и им помага на учениците при изборот на тема за нивната веб-страница, на пример:<br><br>**Лични и образовни теми:**<br><br>1. **Моето училиште** – Презентација на училиштето, наставниците, предметите, активности.<br>2. **Мој дневник / Биографија / Портфолио** – Лична презентација, интереси, фотографии.<br>3. **Мојот омилен предмет** – Интерактивна страна за омилен предмет: географија, хемија итн.<br><br>**Креативни и хоби теми:**<br><br>1. **Моето хоби** – Страница за личен интерес: цртање, музика, танцување, спорт и сл.<br>2. **Гејминг портал** – Презентација на омилени игри, кратки рецензии, галерија. |

3. **Фотографија и дизајн** – Галерија со слики, уредено со CSS flex.

**Општествени и едукативни:**

1. **Заштита на животната средина** – Тема со едукативна содржина и совети.
2. **Здрав начин на живот** – Исхрана, физичка активност, хигиена.
3. **Наука за деца** – Објаснување на научни појави на едноставен начин.
4. **Безбедност на интернет** – Совети за сајбер-безбедност и одговорно користење на интернетот.

**Култура и забавни теми:**

1. **Омилен музички бенд / изведувач** – Биографија, дискографија, видео-клипови.
2. **Омилен филм или серија** – Опис, актери, сцени, постери.
3. **Патување / Туризам** – Презентација на интересни дестинации во Македонија или светот.
4. **Град / Село / Регион** – Страница посветена на родниот крај, историја, култура, традиции.

**Дополнително**

- Друга тема по избор на учениците

### 3) Потсетување на достапните ресурси

Наставникот потсетува на претходните часови, материјалите, HTML шаблони и CSS упатства. Ги охрабрува учениците самостојно да истражуваат и да ги надоградуваат нивните знаења, да ги користат моделите на AI за помош во истражувањето, самоучењето и пишувањето на код (ChatGPT, Gemini, Grok, Copilot итн.)

Им препорачува извори за инспирација и поддршка:

- W3Schools, https://www.w3schools.com/css/default.asp
- Programiz, https://www.programiz.com/css/getting-started
- Css-tricks.com, https://css-tricks.com/snippets/css/a-guide-to-flexbox/#aa-background
- HTML Color Codes, https://htmlcolorcodes.com/
- CSS Gradient, https://cssgradient.io/

- Google Fonts, https://fonts.google.com/
- Font Awesome, https://fontawesome.com/v4/
- Chrome DevTools, https://developer.chrome.com/docs/devtools
- Sublime Text, https://www.sublimetext.com/download
- FreeCodeCamp, https://www.freecodecamp.org/
- MDN Web Docs, https://developer.mozilla.org/en-US/
- Web.dev, https://web.dev/learn/css/welcome
- Webcode.tools, https://webcode.tools/css-generator
- html-css-js.com, https://html-css-js.com/css/
- CodePen, https://codepen.io/pen/

**4) Техничка поддршка  и поттикнување на критичко размислување и самооценување**

Наставникот помага при решавање на конкретни проблеми во кодот (на пр. CSS не се применува, сликите не се вчитуваат, неправилен распоред). Поттикнува критичко размислување преку поставување на прашања: "Зошто ја избра оваа боја? ", "Како можеш да го подобриш изгледот на навигацијата?", "Дали твојата страница е прегледна и читлива?" и слично.

**5) Процес на изработка на проектот**

Учениците избираат тема за веб-страница и ја планираат нејзината структура (воведна страница, мени, секции, слики, текст). Потоа креираат HTML-документ со соодветни елементи и ја стилизираат користејќи CSS. Применуваат знаења за текстуално форматирање, распоредување со „box model“ и користење на класи и идентификатори. Проверуваат дали структурата и дизајнот се функционални и одговараат на избраната тема, ги тестираат линковите, навигацијата, читливоста и уредноста на веб-страницата. Самостојно или во парови ги разгледуваат завршените проекти, даваат повратни информации и го прилагодуваат дизајнот според препораки од наставникот и соучениците.

| | |
|---|---|
| • **Презентација и евалуација на проектот и утврдување на оценките**<br><br>Број на часови: 1 (36) | **Презентација на проектите**<br><br>○ Учениците (во парови или тимови) ја презентираат нивната веб-страница пред одделението;<br>○ Објаснуваат што содржи, што научиле, со кои проблеми се соочиле и како ги решиле.<br><br>**Оценување и повратна информација**<br><br>○ Наставникот ги споделува и објаснува критериумите за оценување;<br>○ Оценува според однапред најавените критериуми (дизајн, примена на знаења, креативност, техничка исправност);<br>○ Дава повратна информација и предлози за подобрување.<br><br>**Рефлексија**<br><br>○ Разговор со учениците:<br><br>1) "Што научивте преку процесот на изработка на оваа страница? "<br>2) "Што би направиле поинаку следниот пат? " |

# СОДРЖИНА