**Begin**

Number : D, E, C.

Print (Evro)

Input E

C = E \* 117

Print (Iznos)

Output C

**IF**

Begin

Ocena 1, ocean 2, ocean 3, prosek

Print (ocean 1)

Print (ocean 2)

Print (ocena 3)

Input ocena 1, ocena 2, ocena 3

Prosek = (ocena 1 + ocena 2 + ocena 3) / 3

If

Prosek > 60 than print Successful

Else

Print failed

END If

End.

**Begin**

Var Mile, kilometer, result

Print (upisi\_iznos)

Input iznos

Print (unesi mernu jedinizu - milje ili kilometre)

Input (merna\_jedinica)

Print (unesi mernu jedinicu u koju konvertujes)

Input ciljana\_merna\_jedinica

If merna\_jedinica = mile and ciljana\_merna\_jedinica = kilometer than

Result = iznos \* 1,0303

Print result

If merna\_jednica = kilometer and ciljana\_merna\_jedinica mile than

Rezultat = iznos \* 0.223

Print rezultat

End.

Convert miles to kilometers or kilometeres to miles

**Begin**

Var numbers (‘amount’, ‘press’, result’)

Print (“press 1 if you want to convert miles to kilometers or press 2 if you want to convert kilometers to miles”)

Input Press

Print (amount)

Input amount

If press = 1 than

Result = amount \* 1,041

Else if press = 2 than

Result = amount \* 0.324

Else Print (something went wrong)

End if.

Print Result

End.

**Arrays, lists - kolone, liste**

Var studenti = [ ‘Maja’ , ‘Mirko , ‘Stefan’ ]

For i=0 ; i<3 ; i=i++

Print studenti [0]

Print studenti [1]

Print studenti [2]

For i=0 ; i<3 ; i=i++

Print I (u ovom slucaju ispisuje samo brojeve(menja brojeve))

Var studenti = [ ‘Maja’ , ‘Mirko , ‘Stefan’ ]

For i=0 ; i<3; i=i++

Print studenti **[i]** (sada ce ispisati imena studenata)

Bilo bi –

Var studenti = [ ‘Maja’ , ‘Mirko , ‘Stefan’ ]

For i=0 ; i<3; i=i++

Print studenti [i]

Var studenti = [ ‘Maja’ , ‘Mirko , ‘Stefan’ , ‘Suzana’ , ‘Djordje’ ]

For i=0 ; **i<studenti.length** ; i=i++ (.length oznacava da se ispisuje do kraja, beskonacno)

Print studenti **[i]**

**Zadatak da se ispise ime jednog od svih studenata**

Begin

Var Studenti [‘Maja’ , ‘Anastazija’ , ‘Djordje’]

Var student\_pronadjen = false ;

Print ‘Unesi ime’

Input ime

For var i=0 ; i< studenti.length ; i = i++

If ime == student[i]

Print ‘ime’

? End program (ako ovo dodamo, ovde se zavrsava niz FOR I ponavljanja) ?

Student\_pronadjen == true

End for

If student\_pronadjen == false

Print ‘ime nije na spisku’

End

Begin

Var – broj1, broj2, broj3, rezultat

Print (upisi\_broj1)

Input broj1

Print (upisi\_broj2)

Input broj2

Print (upisi\_broj3)

Input broj3

Rezultat = broj1 + broj2 + broj3 / 3

Print rezultat

End.

**UVOD – HTML / CSS**

* **ARPANET** - The name of the first computer network (1969)
* **INTERNET** - collection of nodes (computers, routers, networks / networks based on the tci / ip communications protocol)
* **INTERNET** **SERVERS** (Email, WWW)
* **IP** ?
* **WWW** (father of www was Tim Berners-Lee, the universe of network-accessible information, an embodiment of human knowledge)
* **WWW** (technical definition / all the users resources on the internet using HTTP)
* **HTTP** – when clicking a link, you are transferring an URL to you browser, your browser knows which server to contact and what file to ask for.
* **URL** (****Absolute URL, Relative URL****)– fundamental network, used to specify addresses on WWW.
* **WEB SERVERS** – always connected to the internet, every site on it has unique address made up of series of four numbers between 0 and 256 separated by periods.
* **WEB SERVERS** – when you register a web address (domain (your site)), you have to specify the IP address of the web server that will host the site.
* **DOMAIN** – main part of your web site – www.mysite.com – mysite.com is domain.
* **SUBDOMAIN** – further dinision of your domain
* **DOMAIN EXTENSIONS** - .com / .net / .gov
* **WEBSITES** - a group of interlinked and well -structured web pages on the same domain
* **WEB BROWSERS** – Microsoft edge, Mozilla, chrome
* **HOSTING** - ?
* **SEO** – search engine optimization process of improving the quality and quantity of website

**HTML – hyper text markup language**

* Used to format the text, you can add instructions to tell the computer how to display the text
* <please do this> / </end please do this>
* **Tools** for HTML code – Visual Studio, Web flow
* **HTML** consists of an opening tag, content and closing tag.
* <p lang =”fr”> (lang = attribute name, fr = attribute value)
* **STRUCTURE HTML**

<html>

<head>

<title> page title here </title>

<head>

<body>

<body>

</html>

* **CTRL** a = MARKING
* **CTRL** C = COPY
* **CTRL** V = PASTE
* **CTRL** Z = UNDO
* **CTRL** S = SAVE
* **CTRL** / = Comment

**COMENTS** (<! -- comments -->) shortcut = ctrl /

* With comments you can place notifications and reminders in your HTML code:

**Headings**

**<h1>**This is heading 1</h1>

* <h1 style="color:red;">Heading 1</h1>

<h2>This is heading 2</h2>  
<h3>This is heading 3</h3>

**Paragraphs**

**<p>**This is a paragraph</p>

**<hr> (**thematic break**)**

<p>This is some text</p>  
<hr> (Ovo u tekstu ostaje prazno ili oznaceno kao horizontalna linija, nakon hr moze se uvoditi novi H ili P.)

**<br> (**a new line without starting a new paragraph**)**

<p>This is<br>a paragraph<br>with line breaks</p>

**<pre>**

<p>Ispod je pesma koja je izdeljena (pre se najvise koristi za to):</p>

<pre>

My Bonnie lies over the ocean.

My Bonnie lies over the sea.

My Bonnie lies over the ocean.

Oh, bring back my Bonnie to me.

</pre>

**Links** - praktikovati dodavanje TITLE-a

<**a** href="https://www.w3schools.com">This is a link</a>

The link's destination is specified in the href attribute.

Atribut TARGET (\_blank (opens in new window) or \_self (opens in the same window))

<a href="https://www.youtube.com/" target="\_blank"> YOUTUBE </a>

**ANCHOR ELEMENT - bookmark**

Stavlja se **id** u neki tag I na to se vraca kada se klikne na “back”

<body id="top">Ovde se pise kod </body>

<a href="#top">Back to top</a>

**Mailto**

<a href="mailto:jon@example.org">Email Jon</a>

**IMG like link**

<a href="default.asp">

  <img src="smiley.gif" alt="HTML tutorial" style="width:42px;height:42px;">

</a>

**BUTTON like link**

<button onclick="document.location='write a link'">HTML</button>

**Colors of links : link, hover, visited, active**

a:visited {

    color: pink;

    background-color: transparent;

    text-decoration: none;

  }

**Images**

<**img**src="w3schools.jpg" alt="error" width="104" height="142">

- Using **STYLE** in **INLINE CSS** with **IMG** tag

<!DOCTYPE html>

<html>

<head>

<style>

img {

  width: 100%;

}

</style>

</head>

<body>

<img src="html5.gif" alt="HTML5 Icon" width="128" height="128">

<img src="html5.gif" alt="HTML5 Icon" style="width:128px;height:128px;">

</body>

</html>

**Float img**

<p><img src="smiley.gif" alt="/" style="float:right;width:42px;height:42px;">

    The image will float to the right of the text.</p>

**IMAGE MAP**

* rect - defines a rectangular region
* circle - defines a circular region
* poly - defines a polygonal region
* default - defines the entire region

<img src="workplace.jpg" alt="Workplace" usemap="#workmap" width="400" height="379">

<map name="workmap">

  <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm">

  <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">

  <area shape="circle" coords="337,300,44" alt="Cup of coffee" href="coffee.htm">

</map>

**- IMG background and SIZE !!!**

- 10px 10px; width: 10px; height: 10px; - prvih 10px 10px oznacava samo velicinu slike a ne elementa !!!

<p style="background-image: url(anthurium-sch.jpg);

        background-size: 10px 10px; width: 10px; height: 10px; !!!!!!!!

        background-repeat: no-repeat;">

</p>

**-** to make sure the entire element is always covered, set the background-attachment property to **fixed**;

- background-size = **cover**

<style>

    body {

  background-image: url('img\_girl.jpg');

  background-repeat: no-repeat;

  background-attachment: fixed;

  background-size: cover;

}

</style>

- Ako hocemo da slika zauzima uvek ceo element, mzemo staviti **“strech” - background-size 100% 100%**

**<picture> -** The <picture> element contains one or more <source> elements, each referring to different images through the srcset attribute. This way the browser can choose the image that best fits the current view and/or device.

- Bitno je da se **min-width** postavi tako da prvo ide najveci **PX**, jer to ima prioritet

<picture>

    <source media="(min-width: 650px)" srcset="img\_food.jpg">

    <source media="(min-width: 465px)" srcset="img\_car.jpg">

    <img src="img\_girl.jpg" style="width:auto;">

</picture>

**FAVICON image** - Ikona u uglu prozora na pretrazivacu

<head>

<title>Document</title>

    <link type="image/x-icon" rel="icon" href="anthurium-sch.jpg">

</head>

**Empty HTML Elements**

HTML elements with no content are called empty elements.

<p>This is a <br> paragraph with a line break. </p>

**Text éléments**

**<title>** (kad se predje klikom preko reci, pisace dodatni tekst)

**<b>**Bold </b> **BOLD**

**<i>**Italic</i> *ITALIC*

**<u>**Underline</u> UNDERLINE

**Strong** <p><strong>This text is important! </strong></p>

**EM (emphasis)** <p><em>This text is emphasized. </em></p> (This text is emphasized.)

**Small** <p><small>This is some smaller text. </small></p>

**Mark** <p>Do not forget to buy <mark>milk</mark> today. </p> (Do not forget to buy milk today.)

**Del** <p>My favorite color is <del>blue</del> red.</p> (My favorite color is  red.)

**Ins** <p>My favorite color is <del>blue</del> <ins>red</ins>. </p> (My favorite color is  red.)

**SUB** <p>This is <sub>subscripted</sub> text. </p> (This is subscripted text.)

**SUP** <p>This is <sup>superscripted</sup> text. </p> (This is superscripted text.)

**Blockquote**  <blockquote cite="http://www.worldwildlife.org/who/index.html">

<p> For 60 years, WWF has worked to help people and nature thrive. </p>

</blockquote>

<blockquote onclick="alert('Hello!')"> <p>Hello!</p> </blockquote>

**<q>** <p>WWF's goal is to: <q>Build a future where people live in harmony with nature. </q></p>

WWF's goal is to: “Build a future where people live in harmony with nature. “

**<abbr>** <p>The <abbr title="World Health Organization">WHO</abbr> was founded in 1948.</p>

The WHO was founded in 1948.

**<address>**

<address>Written by John Doe.<br>

Visit us at:<br>

Example.com<br>

Box 564, Disneyland<br>

USA

</address>

*Written by John Doe.  
Visit us at:  
Example.com  
Box 564, Disneyland  
USA*

***<cite*>** <p><cite>The Scream</cite> by Edvard Munch. Painted in 1893. </p>

**<bdo>** <bdo dir="rtl">This text will be written from right to left</bdo>

**Lists**

* **Ordered List**

<ol>  
  <li>Coffee</li>  
</ol>

* **Unorderd list**

<ul>  
  <li>Coffee</li>  
</ul>

* **Description list**

<dl>  
  <dt>Coffee</dt>  
   <dd>- black hot drink</dd>  
</dl>

* **Nested lists**

<ul>  
  <li>Coffee</li>  
 <li>Milk</li>  
 <li>Water

<ul>  
 <li> Plain </li>  
 <li> Mineral </li>

</ul>

</li>  
</ul>

**Oznacavanje u listama:**

- **Unordered list** - https://www.w3schools.com/html/html\_lists\_unordered.asp

<ul style="list-style-type:disc;">

    <li>Coffee</li>

    <li>Milk</li>

</ul>

- disc - pun krug

- circle - krug

- square - kockica

- none

- lower-alpha

- upper-roman

**Ordered list** - https://www.w3schools.com/html/html\_lists\_ordered.asp

<ol type="1">  **umesto 1 moze biti, I ili malo i za rimske brojeve, A za slova…**

    <li>Coffee</li>

    <li>Tea</li>

    <li>Milk</li>

</ol>

- moze se dodati atribut start, da bi se oznacilo odakle pocinje brojanje u odredjenoj tabeli

<ol type="1" start="50">

    <li>Coffee</li>

    <li>Tea</li>

    <li>Milk</li>

</ol>

**- list-style-image**

    ul {

        list-style-image: url('sqpurple.gif');

      }

**- list-style-position**: inside, outside

ul.a {

  list-style-position: outside;

}

**- Primer uredjenja liste :**

<https://www.w3schools.com/css/tryit.asp?filename=trycss_list-style_colors>

**TABELE**

 <**table>** - Specifies a table

 <**caption>** - Specifies a table caption

 <**thead>** - Specifies table header row

 <**tbody>** - Specifies table body rows

 <**tfoot>** - Specifies table footer rows

 <**tr>** - Specifies a table row

 <**th>** - Specifies a table header cell

 <**td>** - Specifies table data cell

<table>

<thead>

<tr>

<th colspan="2">The table header</th>

</tr>

</thead>

<tbody>

<tr>

<td>The table body</td>

<td>with two columns</td>

</tr>

</tbody>

<tfoot> </tfoot>

</table>

- colspan - The colspan attribute can be used on a <th> or <td> element and indicates how many columns that cell should run across

- rowspan - The rowspan attribute can be used on a <th> or <td> element to indicate how many rows a cell should span down the table

 <**table border="n">** - ordedjuje debljinu ivice

 <**table cellpadding="n">** - razmak sadrzaja svake celije I njenih ivica

 <**table cellspacing="n">** - razmak izmedju celija

- BORDER

· solid – puna linija (kao u tvom primeru)

· dashed – isprekidana linija

· dotted – tačkasta linija

· double – dupla linija

· groove – gravirana linija koja daje utisak urezivanja

· ridge – suprotno od groove, daje efekat izdizanja

· inset – linija koja izgleda kao da je ugrađena unutar elementa

· outset – linija koja izgleda kao da izlazi iz elementa

· none – bez bordera

- **padding in CSS** = Stilizovanje tabela

<table style="border: 1px dotted red;" >

<table style="border: 1px dotted red;  border-right-width: 10px; border-bottom: 5px double yellow;"  >

- https://www.w3schools.com/html/html\_table\_borders.asp

**Border** collapse !!!

<style>

    table, th, td {

  border: 1px solid black;

  border-collapse: collapse;

}

</style>

**Border** radius

<style>

    table, th, td {

      border: 1px solid black;

      border-radius: 10px;

    }

    </style>

**Table** size

<table style="width:100%">

    <tr>

      <th style="width:70%">Firstname</th>

    </tr>

</table>

- <th style="border: 1px solid black; width: 40%;" >dasdsa</th>

- Height

<th style="border: 1px solid black; width: 40%; **height:** 30px;" >Ime</th>

- Text-align

th {

  text-align: left;

}

- **Table** caption - daje da iznad tabele imamo natpis, tako izbegavamo druge elemente !!!

<table style="width:100%">

        <caption>Monthly savings</caption>

    <tr>

      <th>Month</th>

    </tr>

</table>

**Padding** - razmak izmedju sadrzaja I ivice celije.

th, td {

  padding-top: 10px;

  padding-bottom: 20px;

  padding-left: 30px;

  padding-right: 40px;

}

**Table styling** color of the columns and rows

- <https://www.w3schools.com/html/html_table_styling.asp>

- u style oznacimo th, td ili tr I onda pisemo :  **nth-child(even) : nth-child(odd)**

<style>

    table, th, td {

      border: 1px solid black;

      border-collapse: collapse;

    }

    tr:nth-child(even) {

      background-color: rgba(150, 212, 212, 0.4);

    }

    th:nth-child(even),td:nth-child(even) {

      background-color: rgba(150, 212, 212, 0.4);

    }

</style>

:hover - prilikom prelaska misom preko celija, menja se njihova boja

td:hover{

    background-color: aqua;

}

#nekiId: hover{

    background-color: black;

}

Colgroup - dodavanje stila odredjenim **kolonama**

<table style="width: 400px;">

    <colgroup>

      <col span="2" style="background-color: #D6EEEE">

    </colgroup>

    <tr>

    <th>MON</th>

    <th>TUE</th>

    <th>WED</th>

    </tr>

</table>

- tek od cetvrte kolone dodaje se stil.

    <colgroup>

        <col span="3">

        <col span="2" style="background-color: pink">

    </colgroup>

- prvi col span odredjuje prve kolone. Sledeci col span se odnoso na kolone nakon zadate vrednosti u prvom col span-u !!!

**Primer stilizovanja tabele bez linija:**

table {

  border-collapse: collapse;

  width: 100%;

}

th, td {

  padding: 8px;

  text-align: left;

  border-bottom: 1px solid #ddd;

}

**Responsibe table** : tabela se stavlja u neki div element sa sledecim propertijem

<div style="overflow-x:auto;">

    <table>table content</table>

</div>

**BLOCK ELEMENTS -**  h1, table, ol, ul, nav, hr, pre, form, address, div, p

<aside>

<p>My family and I visited The Epcot center this summer.</p>

<aside>

<p>The Epcot center is a theme park at Walt Disney World Resort.</p>

</aside>

<p>My family and I visited The Epcot center this summer. </p>

- css <aside>

<style>

    aside {

      width: 30%;

      padding-left: 15px;

      margin-left: 15px;

      float: right;

      font-style: italic;

      background-color: lightgray;

    }

</style>

<article> - koristi se za artikle : forum post, blog post, news story

- https://www.w3schools.com/tags/tag\_article.asp

<main> - The content inside the <main> element should be unique to the document. It should not contain any content that is repeated across documents such as sidebars, navigation links, copyright information, site logos, and search forms.

****Note:**** There must not be more than one <main> element in a document. The <main> element must NOT be a descendant of an <article>, <aside>, <footer>, <header>, or <nav> element.

<footer> - A <footer> element typically contains:

* authorship information
* copyright information
* contact information
* sitemap
* back to top links
* related documents

You can have several <footer> elements in one document.

<video> - Atributi video taga su :

**- type** (video/mp4 ... video/ogg)

- autoplay

- poster (url slike koja ce biti poster na videu)

- loop

- controls

- preload (auto(upload whole video when page load), metadata(upload only metadata info), none(no loading video when the page load))

- muted - audio is muted

- Ove atribute mozemo dodavati na kraj URL-a **IFRAME**:

<iframe width="420" src="https://www.youtube.com/embed/tgbNymZ7vqY?controls=1"></iframe>

**Controls** : 0 - nevidljive kontrole, 1 - vidljive

**Loop** : 0 - loop jednom; 1 - forever. Ako se dodaje loop, onda se mora dodati i **playlist** i njemu dati vrednost ID-a videa da bi loop mogao da se obavlja.

**Autoplay** : 1 - krece sam

**Muted** : 1 - mutiran

- ovi atributi se spajaju sa &

- **source** tag se koristi kada imamo vise formata videa da bi se ucitao podrzani video

<video width="320" height="240" controls>

    <source src="movie.mp4" type="video/mp4">

    <source src="movie.ogg" type="video/ogg">

</video>

**INLINE ELEMENTS**

<script> - The <script> tag is used to embed a client-side script (JavaScript). The <script> element either contains scripting statements, or it points to an external script file through the src attribute.

    <p id="demo"></p>

    <script>

    document.getElementById("demo").innerHTML = "Hello JavaScript!";

    </script>

<cite> - opisuje ime kreativnog rada. Ime knjige, umetnickog dela…

<img src="img\_the\_scream.jpg" width="220" height="277" alt="The Scream">

    <p><cite>The Scream</cite> by Edward Munch. Painted in 1893.</p>

<span> - slicno div (block element), sluzi za deo teksa ili dokumenta

<p>My mother has <span style="color:blue">blue</span> eyes.</p>

**Inline elements** : i, em, img, map, small, strong, b, a, br, sub, sup, input, textarea, q

<iframe> - za ubacivanje sadrzaja sa druge url stranice, cesto za google map ili youtube video

<iframe src="demo\_iframe.htm" style="height:200px;width:300px"

title="Iframe Example"></iframe>

<iframe src="https://www.youtube.com/embed/5U6QPCYlqQQ" title="video"

        allow="accelerometer; - promena brzine I orijentacije

               autoplay;

               clipboard-write;

               encrypted-media;

               gyroscope; - promena pokreta I naginjanja

               picture-in-picture"

        frameborder="0"

        allowfullscreen>

</iframe>

- Kada koristiš <iframe> za ugradnju YouTube videa, moraš koristiti **embed** URL, a ne običan URL za gledanje videa (**watch?v=**). Umesto toga se pise **embed/**

## - Iframe - Target for a Link

## - Atribut name u iframe I atribut target u a(linku) moraju da se poklapaju da bi to funkcionisalo

<iframe src="demo\_iframe.htm" name="iframe\_a" title="Iframe Example"></iframe>

<p><a href="https://www.w3schools.com" target="iframe\_a">W3Schools.com</a></p>

<p>When the target attribute of a link matches the name of an iframe, the link will open in the iframe.</p>

**Iframe za GIF** - na sajtu nadjes gif al uzimas embed opciju

<iframe src="https://giphy.com/embed/n7M8hEMvRRkjBfQb8k" width="400" height="480"  frameBorder="0" class="giphy-embed" allowFullScreen></iframe><p><a href="https://giphy.com/gifs/n7M8hEMvRRkjBfQb8k">via GIPHY</a></p>

### Razlika između **inline** i **block** elemenata:

#### 1. ****Block elementi****:

* **Primeri**: <div>, <p>, <h1>, <section>, <article>, <table>
* **Karakteristike**:
  + Zauzimaju **celu širinu** dostupnog prostora (od jednog kraja stranice do drugog), čak i ako njihov sadržaj ne ispuni celu širinu.
  + Svaki block element počinje **novi red**.
  + Možeš postavljati veličinu (**width**, **height**) i margine oko block elemenata.

#### 2. ****Inline elementi****:

* **Primeri**: <span>, <a>, <img>, <strong>, <em>, <input>
* **Karakteristike**:
  + Zauzimaju samo onoliko prostora koliko je potrebno za njihov sadržaj.
  + **Ne počinju novi red** i mogu biti postavljeni jedan pored drugog.
  + **Ne možeš** im postaviti širinu ili visinu (osim kod posebnih inline elemenata poput <img>).
  + Margine i padding deluju samo horizontalno, ali ne i vertikalno.

### Zašto je ovo važno za pozicioniranje?

* **Block elementi** se koriste kada želiš da element zauzme ceo prostor (npr. paragrafi, naslovi, divovi koji grupišu sadržaj).
* **Inline elementi** se koriste za manje delove sadržaja koji treba da se pojavljuju unutar linije teksta (npr. linkovi ili stilizacija unutar paragrafa).

### Pozicioniranje:

**Block elementi**:

* + Lako ih je pozicionirati pomoću **margin**, **padding**, i **position** svojstava (kao što su relative, absolute, ili fixed).
  + Možeš im podesiti veličinu i položaj bez problema jer zauzimaju blokovski prostor.

**Inline elementi**:

* + Teže je primeniti width i height direktno na njih, jer zauzimaju samo prostor potreban za njihov sadržaj.
  + Postavljanje inline elemenata je često vezano za kontekst u kojem se nalaze (npr. unutar paragrafa teksta).

**HEAD** part of the document

**- <title>**  tag is required. The content of a page title is very important for search engine optimization (SEO)! The page title is used by search engine algorithms to decide the order when listing pages in search results.

**<script>** - JS

**<link>** - inport CSS and favicon icon

**<style>** - CSS

**<base>** - all relative URL on the page, (svi linkovi na stranici ce se ponasati na isti nacin)

<head>

    <base href="https://www.w3schools.com/" target="\_blank">

</head>

**<meta>** - The <meta> element is typically used to specify the character set, page description, keywords, author of the document, and viewport settings.

The metadata will not be displayed on the page, but is used by browsers (how to display content or reload page), by search engines (keywords), and other web services.

<head>

    <meta charset="UTF-8"> !!!

    <meta name="description" content="Free Web tutorials">

    <meta name="keywords" content="HTML, CSS, JavaScript">

    <meta name="author" content="John Doe">

<meta http-equiv="refresh" content="30"> - ucitavanje stranice svakih 30 sec.

<meta name="viewport" content="width=device-width, initial-scale=1.0">  
…………<meta http-equiv="expires" content="Fri, 04 Apr 2014 23:59:59 GMT" />  
… <meta http-equiv="pragma" content="no-cache" /> nema kes podataka

</head>

**Setting the viewport**

The width=device-width part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).

The initial-scale=1.0 part sets the initial zoom level when the page is first loaded by the browser.

Responsive web design

- Responsive Web Design is about using HTML and CSS to automatically resize, hide, shrink, or enlarge, a website, to make it look good on all devices (desktops, tablets, and phones):

- **Viewport =** osnovno

- **Size** = responsive image, mozemo podestiti width na 100% ali ona onda moze biti veca od originala, zato stavljamo npr max-width = 100%

**<picture>** = ovaj element, tag, naucili smo ga ranije !

**Text-size** = velicinu teksta postaviti na **vw** (viewport width)

- Viewport is the browser window size. 1vw = 1% of viewport width. If the viewport is 50cm wide, 1vw is 0.5cm.

<h1 style="font-size:10vw">Hello World</h1>

**@media** = da prikaze skroz drugaciji stil za razlicite velicine ekrana

<style>

/\* Use a media query to add a break point at 800px: \*/

@media screen and (max-width: 800px) {

  .left, .main, .right {

    width: 100%; /\* The width is 100%, when the viewport is 800px or smaller \*/

  }

}

</style>

Full exemple

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<style>

\* {

  box-sizing: border-box;

}

.menu {

  float: left;

  width: 20%;

  text-align: center;

}

.menu a {

  background-color: #e5e5e5;

  padding: 8px;

  margin-top: 7px;

  display: block;

  width: 100%;

  color: black;

}

.main {

  float: left;

  width: 60%;

  padding: 0 20px;

}

.right {

  background-color: #e5e5e5;

  float: left;

  width: 20%;

  padding: 15px;

  margin-top: 7px;

  text-align: center;

}

@media only screen and (max-width: 620px) {

  /\* For mobile phones: \*/

  .menu, .main, .right {

    width: 100%;

  }

}

</style>

</head>

<body style="font-family:Verdana;color:#aaaaaa;">

<div style="background-color:#e5e5e5;padding:15px;text-align:center;">

  <h1>Hello World</h1>

</div>

<div style="overflow:auto">

  <div class="menu">

    <a href="#">Link 1</a>

    <a href="#">Link 2</a>

    <a href="#">Link 3</a>

    <a href="#">Link 4</a>

  </div>

  <div class="main">

    <h2>Lorum Ipsum</h2>

    <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>

  </div>

  <div class="right">

    <h2>About</h2>

    <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit.</p>

  </div>

</div>

<div style="background-color:#e5e5e5;text-align:center;padding:10px;margin-top:7px;">© copyright w3schools.com</div>

</body>

</html>

**Computer code**

- https://www.w3schools.com/html/html\_computercode\_elements.asp

Tagovi <kbd> (input) and <samp> (output)

**URL**

A URL is another word for a web address.

A URL can be composed of words (e.g. w3schools.com), or an Internet Protocol (IP) address (e.g. 192.68.20.50).

Web browsers request pages from web servers by using a URL.

A web address like https://www.w3schools.com/html/default.asp follows these syntax rules:

scheme: //prefix. domain: port/ path/ filename

* **Scheme (skim)** - defines the **type** of Internet service (most common is **http or https (secured)**)
* ****prefix**** - defines a domain **prefix** (default for http is **www**)
* **domain** - defines the Internet **domain name**(like w3schools.com)
* **port** - defines the **port number**at the host (default for http is **80**)
* **path** - defines a **path** at the server (If omitted: the root directory of the site)
* **filename** - defines the name of a document or resource

**XHTML - https://www.w3schools.com/html/html\_xhtml.asp**

**FORM**

· **GET**: Podaci se šalju kao deo URL-a, što znači da su vidljivi u adresnoj traci (npr., example.com/form?name=John&age=25). Obično se koristi za preuzimanje podataka ili kada nije bitno ako su podaci javno vidljivi. GET ima ograničenje veličine (oko 2000 karaktera).

· · **POST**: Podaci se šalju u telu HTTP zahteva, što ih čini nevidljivim u URL-u. POST je pogodan za slanje velikih količina podataka ili poverljivih informacija (poput lozinki), jer se ne prikazuju u adresnoj traci i nemaju ograničenje veličine.

·

-Input koji se ponasaju kao submit: **submit, reset, button, image,**

**BUTTON** - Ako se stavi onclick, ona se ne ponasa kao submit button, ako se ne stavi, onda ce biti submit za tu formu !!!

- Slika kao link, otvara se **A** tag i u njega se ubacuje **IMG** tag !!!

|  |  |
| --- | --- |
| checked | Specifies that an input field should be pre-selected when the page loads (for type="checkbox" or type="radio") |
| disabled | Specifies that an input field should be disabled |
| max | Specifies the maximum value for an input field |
| maxlength | Specifies the maximum number of character for an input field |
| min | Specifies the minimum value for an input field |
| pattern | Specifies a regular expression to check the input value against |
| readonly | Specifies that an input field is read only (cannot be changed) |
| required | Specifies that an input field is required (must be filled out) |
| size | Specifies the width (in characters) of an input field |
| step | Specifies the legal number intervals for an input field |
| value | Specifies the default value for an input field |

- **form attribute** - <https://www.w3schools.com/html/html_forms_attributes.asp>

<form action="https://www.youtube.com/watch?v=CKTsSqda5O4" method="get" target="framename">

    <!-- Text -->

    <label for="text">Ime i Prezime</label>

    <input type="text" id="text" name="text" value="Npr.Marko">

    <br>

    <!-- Password -->

    <label for="password">Password</label>

<input type="password" id="password" name="password" value="Sifra">

     <!-- Radio button -->

    <input type="radio" id="ime" name="ime" value="ime" checked>

    <label for="ime"> Ime </label><br>

    <input type="radio" id="prezime" name="ime" value="prezime">

<label for="prezime">Prezime</label>

<!-- da bi mogao da cekiras samo jednu opciju, NAME mora da bude isto svuda -->

<br><br>

<!-- CheckBox -->

<input type="checkbox" id="soba" name="soba" value="soba" checked>

<label for="soba"> SOba</label>

<input type="checkbox" id="soba1" name="soba1" value="soba1">

<label for="soba1">Sobaaaaaaaaa</label>

<!-- Button ... kao submit button -->

<input type="button" onclick="alert('Caooooo')" value="klikni me"><br><br><br><br>

<!-- color -->

<label for="color">Your fav color?</label>

<input type="color" id="color" name="color" value="boja...">

<!-- date -->

<label for="bday">Your B day</label>

<input type="date" id="bday" name="bday">

<!-- datetime-local -->

<label for="birthdaytime">Birthday (date and time):</label>

<input type="datetime-local" id="birthdaytime" name="birthdaytime"> <br><br><br><br>

<!-- email -->

<label for="email">Enter your email:</label>

<input type="email" id="email" name="email">

<!-- image ... radi kao submit button -->

<input type="image" src="img\_submit.gif" alt="Submit" width="48" height="48">

<!-- file -->

<label for="myfile">Select a file:</label>

<input type="file" id="myfile" name="myfile">

<!-- month -->

<label for="bdaymonth">Birthday (month and year):</label>

<input type="month" id="bdaymonth" name="bdaymonth">

<!-- number -->

<label for="quantity">Quantity (between 1 and 5):</label>

<input type="number" id="quantity" name="quantity" min="1" max="5">

<!-- range -->

<label for="vol">Volume (between 0 and 50):</label>

<input type="range" id="vol" name="vol" min="0" max="50" value="10">

<!-- search -->

<label for="gsearch">Search Google:</label>

<input type="search" id="gsearch" name="gsearch">

<!-- tel -->

<label for="phone">Enter a phone number:</label><br><br>

<input type="tel" id="phone" name="phone" placeholder="123-45-678" pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}" required><br><br>

<title>Format: 123-45-678</title><br><br>

<input type="submit" value="Submit">

<!-- time -->

<label for="appt">Select a time:</label>

<input type="time" id="appt" name="appt">

<!-- url -->

<label for="homepage">Add your homepage:</label>

<input type="url" id="homepage" name="homepage">

     <!-- Submit -->

     <input type="submit" value="Zavrsi formu">

     <!-- Reset -->

      <input type="reset" value="Resetuuuuuuuuuhj">

<!-- select --!>

... moze se dodati atribut size (koliko se opcija prikazuje) -->

<!--  ili atribut    (multiple)   -->

<label for="cars">Izaberi auto</label>

<select id="cars" name="cars" required size=”2” multiple>

    <option value="volvo">Volvo</option>

    <option value="peugeot">Peugeot</option>

    <option value="renault">Renault</option>

</select>

<!-- textarea -->

<textarea name="message" rows="10" cols="30">

    The cat was playing in the garden.

</textarea>

<!-- fieldset -->

<fieldset>

    <legend>Personalia:</legend>

    <label for="fname">First name:</label><br>

    <input type="text" id="fname" name="fname" value="John"><br>

    <label for="lname">Last name:</label><br>

    <input type="text" id="lname" name="lname" value="Doe"><br><br>

    <input type="submit" value="Submit">

  </fieldset>

  <!-- output !!! -->

  <form action="/action\_page.php"

  oninput="x.value=parseInt(a.value)+parseInt(b.value)">

  0

  <input type="range"  id="a" name="a" value="50">

  100 +

  <input type="number" id="b" name="b" value="50">

  =

  <output name="x" for="a b"></output>

  <br><br>

  <input type="submit">

</form>

<!-- patern  ... u ovom slucaju, samo 3 slova se mogu upisati-->

<input type="text" id="country\_code" name="country\_code"

        pattern="[A-Za-z]{3}" title="Three letter country code">

<!-- step -->

<input type="number" id="points" name="points" step="3">

<!-- height, width -->

<input type="image" src="img\_submit.gif" alt="Submit" width="48" height="48">

</form>

- **Atributi** : size, max, min, pattern, miltiple, required, readonly, disabled, maxlength (za PIN), value, placeholder, step (kod brojeva)

The placeholder attribute works with the following input types: text, search, url, number, tel, email, and password.

**Value** se ne brise prilikom klika, **placeholder** se brise !!!

**Pattern**

**- samo brojevi**

<input type="text" pattern="\d+" title="Dozvoljeni su samo brojevi">

**- validacija e-mail adrese**

<input type="text" pattern="^[a-z0-9.\_%+-]+@[a-z0-9.-]+\.[a-z]{2,}$" title="Unesite validnu email adresu">

**- lozinka** = najmanje 8 karaktera, barem jedno slovo I jedan broj

(?=.\*[A-Za-z]) - **lookahead** izraz, što znači da postavlja uslov bez uzimanja karaktera iz stringa

<input type="password" pattern="^(?=.\*[A-Za-z])(?=.\*\d)[A-Za-z\d]{8,}$" title="Lozinka mora imati najmanje 8 karaktera, uključujući barem jedno slovo i jedan broj">

**E mail** pattern="[a-z0-9.\_%+-]+@[a-z0-9.-]+\.[a-z]{2,}$":

* [a-z0-9.\_%+-]+: Omogućava mala slova, brojeve i specijalne znakove poput tačke, donje crte itd. za lokalni deo emaila.
* @: Proverava da postoji simbol **@**.
* [a-z0-9.-]+: Omogućava slova, brojeve i tačku za domen (npr. gmail.com).
* \.[a-z]{2,}$: Zahteva da domen ima završetak sa barem dva slova (kao što je .com ili .org).

Pattern uvek ide uz title

- primer za URL :

<input type="url" id="website" name="website" pattern="https?://.+" title="Include http://">

- Samo tri slova :  pattern="[A-Za-z]{3}"

- 8 ili vise karaktera za password :  pattern=".{8,}" title="Eight or more characters"

- Password :   pattern="(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z]).{8,}" title="At least one  number, one uppercase and lowercase letter, and at least 8 or more characters">

- E-mail : pattern="[a-z0-9.\_%+\-]+@[a-z0-9.\-]+\.[a-z]{2,}$"> (characters followed by an @ sign, followed by more characters, and then a "." After the "." sign, add at least 2 letters from a to z)

**[a-z0-9.\_%+\-]+@[a-z0-9.\-]+\.[a-z]{2,}$ - regularni prikaz**

#### 1. [a-z0-9.\_%+\-]+

[a-z0-9.\_%+\-]: Ovo je skup karaktera koji su dozvoljeni u prvom delu email adrese (pre **@** znaka).

* + a-z: Mala slova od a do z.
  + 0-9: Brojevi od 0 do 9.
  + . **(tačka)**: Tačka je dozvoljena u email adresama.
  + \_ **(donja crta)**: Donja crta je dozvoljena.
  + %: Procenat je dozvoljen.
  + +: Plus znak je dozvoljen.
  + \-: Minus znak je dozvoljen (primetite da je znak minus **escaped** pomoću \ jer bi inače imao poseban značaj u regularnim izrazima).

+: Ova oznaka znači da prethodni skup karaktera (sve navedeno iznad) mora da se ponovi **barem jednom**. Dakle, u email adresi pre znaka **@** mora biti barem jedan dozvoljeni karakter.

#### 2. @

* Ovaj znak **@** je statički deo email adrese, i proverava se da li je tačno jedan **@** znak prisutan između korisničkog imena i domena.

#### 3. [a-z0-9.\-]+

* [a-z0-9.\-]: Ovo je skup karaktera koji su dozvoljeni u delu email adrese nakon **@** znaka, a to je domen (npr. **gmail.com**).
  + a-z: Mala slova od a do z u domenu.
  + 0-9: Brojevi od 0 do 9 u domenu.
  + . **(tačka)**: Tačka je dozvoljena za domen (kao u gmail.com).
  + \- **(minus)**: Znak minus je dozvoljen u domenu (npr. my-domain.com).
* +: Kao i pre, znači da prethodni skup karaktera mora da se ponovi **barem jednom**. Dakle, domen mora imati barem jedan karakter.

#### 4. \.[a-z]{2,}$

* \.: Ovo traži **tačku** koja odvaja ime domena od TLD (Top-Level Domain, npr. .com, .org).
* [a-z]{2,}: Ovde se zahteva da TLD (Top-Level Domain) bude sačinjen od **barem dva mala slova** (npr. .com, .org, .net).
  + [a-z]: Mala slova od a do z.
  + {2,}: Ova oznaka znači da mora biti **najmanje dva karaktera**. Tako da su kraći TLD-ovi, kao što su .com, .org, ili .net, validni.

#### 5. $

* $: Ovo označava **kraj niza**. Osigurava da ništa ne dolazi posle TLD-a (npr. korisnik ne može uneti nešto kao example@gmail.com123).

Validacija forme kroz javascript

<input id="numb">

<button type="button" onclick="myFunction()">Submit</button>

<p id="demo"></p>

<script>

function myFunction() {

  // Get the value of the input field with id="numb"

  let x = document.getElementById("numb").value;

  // If x is Not a Number or less than one or greater than 10

  let text;

  if (isNaN(x) || x < 1 || x > 10) {

    text = "Input not valid";

  } else {

    text = "Input OK";

  }

  document.getElementById("demo").innerHTML = text;

}

</script>

Validation:

- input **type**="url" **name**="website" **required pattern**="https**? ://.+**">

- <input **type**="tel" **id**="phone" **name**="phone" **placeholder**="123-45-678" **pattern**="[0-9]{3}-[0-9]{2}-[0-9]{3}" **required**>

**INPUT FORM ATTRIBUTES**

- **formaction** (on ignorise action u form delu. Moze biti korisno npr za : continue as guest). Ide uvek uz submit ili image.

- **formenctype** attribute specifies how the form-data should be encoded when submitted (only for forms with method="post").

- **formmethod** - ignorise metod u form delu. Tu se moze promeniti na post ili get. (mozda korisno kada hocemo da ponudimo klijentu kako da sacuva svoje info iz forme.)

- **formtarget** - prikazuje gde da se otvori rezultat kada se submituje forma.]

- **formnovalidate** - sprecava da se validira forma. Tj. bice poslata iako podaci nisu prethodno provereni. To se desava u situacijama kada kroz JS vrsimo validaciju ili nam nije bitno sta je klijent uneo. Atribut koji nadjacava je atribut novalidate koji se nalazi u **form** delu i sluzi samo za submit dugme !!!

<form action="/action\_page.php">

    <input type="submit" formaction="/action\_page2.php" value="Submit as Admin">

</form>

- **Biblioteka** - koriste se odredjene metode I funkcije koje mozemo pozvati I koristiti.

- **Framework** - Cesto utice na tok aplikacije, ima pravila koje moramo pratiti I definise strukturu aplikacije. (frontend - React, Angular). Za izradu mobilnih aplikacija - React Native (JavaScript), desktop aplikacija - Electron (JavaScript),

**LOKACIJA u HTML-u**

<button onclick="getLocation()">Try It</button>

<p id="demo"></p>

<script>

const x = document.getElementById("demo");

function showPosition(postion){

    x.innerHTML = "Lokacija je sledeca :" + postion.coords.latitude + "<br>"

                                          + postion.coords.longitude;

}

function showError(error){

    switch(error.code){

        case error.PERMISSION\_DENIED:

            x.innerHTML = "Denied"

        case error.POSITION\_UNAVAILABLE:

            x.innerHTML = "Unavailable"

            case error.TIMEOUT:

                x.innerHTML = "Timeout"

            case error.UNKNOWN\_ERROR:

                x.innerHTML = "Unknown error"

                break;

    }

}

function getLocation() {

    if(navigator.geolocation){

        navigator.geolocation.getCurrentPosition(showPosition, showError);

    } else {

        x.innerHTML = "Location is not supported"

    }

}

  </script>

**DRAG and DROP** - css je u style head

<div id="div1" ondrop="drop(event)" ondragover="allowDrop(event)"></div>

<br>

<img id="drag1" src="icon.png" draggable="true" ondragstart="drag(event)">

<script>

    function drag(ev){

        ev.dataTransfer.setData("text", ev.target.id)

}

    function drop(ev){

        ev.preventDefault();

        var data = ev.dataTransfer.getData("text");

        ev.target.appendChild(document.getElementById(data));

}

    function allowDrop(ev){

        ev.preventDefault();

}

</script>

**GIFS**

<img src="https://i.giphy.com/media/v1.Y2lkPTc5MGI3NjExZ3JnOGVqZzA2cWR2dmo0aTg3d2tuZWk5N3pkajhiZ3Rjb2hrMGR5YyZlcD12MV9pbnRlcm5hbF9naWZfYnlfaWQmY3Q9dg/1u01IRKm3cKUH4GU1U/giphy.gif" alt="this slowpoke moves"  width="250" />

**Tag marquee** - moving text

- Modern coding = For creating scrolling effects in modern web development, you can use CSS animations or JavaScript libraries like jQuery.

<marquee behavior="scroll"

         direction="up"

         scrollamount="3"

         loop="-1"

         hspace="5"

         vspace="5">

        - This text scrolls from left to right.

         </marquee>

<!-- Dijagonalno = mora se uraditi nasted marquee -->

<marquee behavior="scroll" direction="right"><marquee style="width:80px; height:100px;" behavior="scroll" direction="up">Text scrolling Diagonally</marquee></marquee>

<!-- Bouncing back and forth -->

<marquee behavior="alternate" scrollamount = "15">Bouncing text</marquee>

Well, almost a year ago i decided to completely change my career. I've been thinking about it for years but i couldn't menage to start with this big learning process. I was looking for a job that would require me to think and improve constantly so I decided to devote my free time to exploring the IT world and i realized that it was a good choice because i feel good while i am doing this, i don't feel stress and I want to become professional in IT development.

Internship in JetBrains seems interesting to me because i think it can help me to improve my knowledge.

The chance to learn from and contribute to a team of experts in fields like software development would greatly enhance my technical skills and professional growth. I also admire the company's commitment to innovation and creating tools that make developers' lives easier. Also, if I get a chance for this internship, I am sure that I will make maximum progress and that with my work i will be able to thank you for choosing me, by doing a great job for the company.

… The programming languages ​​I am good at are HTML, CSS and JAVASCRIPT. I also have some knowledge of C# language, SQL database, but I would like to concentrate on the frontend part so that I can slowly learn in more detail and really become a pro at it.

I worked on a weather forecast site where users could enter a city, and the site would display the current weather for that location. I used HTML and CSS to create a clean and responsive layout, and for the functionality, I used JavaScript along with jQuery to handle the weather data. I also implemented form validation to ensure that users entered valid input. This project helped me get a better understanding of using JavaScript for dynamic content and improve my CSS skills. Beyond this project, I’ve gained experience using jQuery in other contexts, where I've applied it for DOM manipulation, event handling, and simplifying JavaScript tasks, which has helped me become comfortable with the library.

I would like to improve JavaScript very well, but also to learn C# and SQL better, but I know that it cannot be done all at once and that it takes time.

… One of the problems I had was learning CSS positioning. I initially struggled with using CSS Grid and Flexbox to create responsive designs. To solve it, I dedicated time to studying various resources and experimenting with different layout techniques in practice projects. Of course, there are still a lot of things I need to work on when it comes to JavaScript, the programming language I want to perfect. There were problems to understand how and when to use the functions, but through practice and time I managed to learn them better.

**CSS - Cascading Style Sheets**

* CSS stands for Cascading Style Sheets
* CSS describes how HTML elements are to be displayed on screen, paper, or in other media
* CSS saves a lot of work. It can control the layout of multiple web pages all at once
* External stylesheets are stored in CSS files

**Selectors**:

\* {

        color: rgb(128, 2, 2);

}

    h1 {

        color: antiquewhite;

    }

    #paragraph{

        font-size: larger;

    }

    .aside {

        float: right;

        width: 30%;

        background-color: aqua;

        padding: 30px;

    }

    /\* p.fake{    -- to je bio treci P, kada mu se doda aside fake klasa (class=”aside fake”)

        font-style: italic;

        background-color: blue;

} \*/

    aside p {        /\* targetira sve P elemente u aside elementu \*/

        font-size: 10px;

    }

    li > a {       /\* targetira sve A elemente u li elementu - moze bez znaka > \*/

        font-size: 10px;

}

   h1 p {       /\* targetira prvi p element nakon svakog h1 elementa \*/

        background-color: antiquewhite;

    }

</style>

<link rel="stylesheet" href="work2.css">     <!-- external CSS -->

div p - selektuje sve p elemente u div elementu, iako oni mogu biti ugnjezdeni u drugi element koji se nalazi u divu.

div > p - selektuje samo direktnu decu u divu, a ugnjezdene NE.

div + p - selektuje P posle div a.

p:first-child i : selektuje samo prvi p I sve i elemente u tom p.

- important - it overrides all other styles

    p {

       color: red !important;

    }

<h1 class="center">This heading will not be affected</h1>

<p class="center">This paragraph will be red.</p>

<p class="center large">This paragraph will be center-aligned.</p>

***- The last style defined in head section will have priority !!! Inline style has priority in any case.***

 Some properties are not **inherited**

 E.g. background-color or border properties

 Explicitly saying that some element will *inherit* values from their parent elements by using *inherit* for the value of the properties.

**Colors:**

**- RGB values** (rgb (red, green,blue) (100,100,90)) - Each parameter (red, green, and blue) defines the intensity of the color between 0 and 255. BLACK rgb(0, 0, 0).

- **HEX codes** (These are six-digit codes that represent the amount of red, green and blue in a color, preceded by a pound or hash # sign. For example: #ee3e80).

**- HSL** - Hue, Saturation, Lightness. Hue is a degree on the color wheel from *0 to 360. 0 is red, 120 is green, and 240 is blue.*

**- RGBA** - The alpha parameter is a number between 0.0 (fully transparent) and 1.0 (not transparent at all): - (*applies only on background not on the text)*

**- OPACITY -**  always between 0.0 and 1.0 -  *(transparency applies on text and background also)*

Razlika izmedju opacity I RGBA :

https://www.w3schools.com/css/tryit.asp?filename=trycss\_background\_opacity2

**Box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);**

**0px** - horizontalno pomeranje senke od leva na desno

**8px** - vertikalno od gore na dole

**16px** - Radijus zamućenja senke. Što je veći broj, to je senka zamućenija.

**0px** - Radijus širenja senke, koji dodatno povećava senku oko elementa. Pošto je 0, ne utiče na širinu.

**Rgba** - boja

**Border-style**

       border-style: dotted solid groove dashed;

U ovom slucaju ide s leva na desno.

Ako ima 3 vrednosti = 1. top ; 2. left and right ; 3. bottom

Ako ima 2 vrednosti = 1. top bottom ; 2. left right

**Background-image**

body {

        background-image: url(anthurium-sch.jpg);

        background-repeat: no-repeat;

        background-position: right top;

        background-position: 50% 50%;      /\* sa strana \*/

        background-position: 50px 150px;  /\* sa leve strane i 150 px od vrha  \*/

        background-size: 10px 10px; width: 30px; height: 30px; // contain or cover also

        background-attachment: scroll;

        background: red      /\* Shorthand \*/

                    url(anthurium-sch.jpg)

                    no-repeat

                    scroll

                    center

        ;

    }

**- 3 slike (Moze se ubaciti vise slika, sve posebno ili mozemo koristiti skracenice !!! )**

#example1 {

  background-image: url(img\_flwr.gif), url(paper.gif);

  background-position: right bottom, left top;

  background-repeat: no-repeat, repeat;

  padding: 15px;

}

- ili skracenice

   #example1 {

  background: url(img\_tree.gif) left top no-repeat, url(img\_flwr.gif) right bottom no-repeat, url(paper.gif) left top repeat;

  padding: 15px;

  background-size: 50px, 130px, auto;

  }

**background-origin** : padding-box(default) - content-box - border-box

**background-clip:** koristi se kada postavljamo boju za pozadini. Ima iste vrednosti kao origin.

**background-color** : transparent, currentcolor, inherit

**Margin -**

- margin: auto (to horizontally center an block element like <div>) (Postavlja element na sredinu a prostor okolo je jednak sa svih strana).

- text-align (za pozicioniranje teksta ili inline elemenata)

- slika center

img {

  display: block;

  margin-left: auto;

  margin-right: auto;

  width: 40%;

}

- padding: 30px 0; ( vertical-align)

- za poravnjavanje sa horizontalno I vertikalno

.center {

  padding: 70px 0;

  border: 3px solid green;

  text-align: center;

}

- line-height = sa istom vrednoscu height, postavljam tekst na sredinu

: https://www.w3schools.com/css/css\_align.asp

**Padding**

- Dodavanjem box-sizing, ostaje nam zeljena velicina, jer width ne utice na velicinu elementa.

    div {

  width: 300px;

  padding: 25px;

  box-sizing: border-box;

}

**Box-sizing** : **border-box** ili **content-box**

- <https://www.w3schools.com/css/tryit.asp?filename=trycss_padding_width2>

****Note:**** Remember that the height and width properties do not include padding, borders, or margins! They set the height/width of the area inside the padding, border, and margin of the element!

**max-width**

- najveca velicina elementa, ako se smanji velicina ekrana, tekst ostaje I ispisuje se ispod. Ako bi stavili samo width, tekst bi nestajao, ne bi se spustao ispod. (*min-widht, max-height, min-height*)

**Outline** - moze imati sledece propertije:

* - outline-style
* outline-color
* outline-width
* outline-offset - podesava prostor izmedju bordera I outline
* outline

- off-set property podesava prost

**TEXT**

1. **alignment**

- text-align se koristi za poravnanje teksta i inline elemenata unutar jednog block elementa, ali ne može direktno pozicionirati dva bloka (kao što su dva <div> elementa) jedan u odnosu na drugi.

* text-align - used to set the horizontal alignment of a text.
* text-align-last -  specifies how to align the last line of a text.
* direction - used to change the text direction of an element
* unicode-bidi - used to change the text direction of an element -
* vertical-align - vertical alignment (**baseline, text-top, text-bottom, sub, super, middle**)

- When the text-align property is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers). Popunjava celu liniju tekstom I sam odredjuje razmak izmedju reci.

1. **decoration**

* text-decoration-line - (overline / underline / line-trough /, overline underline / none / overline underline line-trough)
* text-decoration-color - (purple, green, red, blue…)
* text-decoration-style - (dotted, solid, dashed, grouve)
* text-decoration-thickness - (debljina, 5px, 25%)

- Linkovi su uvek prikazani sa linijom ispod. Da bi to skinuli pisem line = **NONE** !

1. **text-transform** - (uppercase, lowercase, capitalize )

**4. Text-spacing**

**- line-height** - koristi se da se odredi razmak izmedju linija teksta

**- letter-spacing** - razmak izmedju slova

**- word-spacing** - razmak izmedju reci

**- text-indent**  - uvod prvog pasusa !!!

**5. Text-shadow**

**- text-shadow**: 2px 2px 5px red; (horizontal, vertical, blur effect, color)

Try => text-shadow: 1px 1px 2px black, 0 0 25px blue, 0 0 5px darkblue;

1. **Text-overflow**

text-overflow: clip;

text-overflow: ellipsis; (ima ttri tacke koje pokazuju da postoji jos teksta)

  overflow: hidden;  
  text-overflow: ellipsis;

1. **Word-wrap** - dozvoljava da se dugacka rec presece da bi isla u novi red

  word-wrap: break-word;

1. **Word-break** - line braking rule. Na kojoj liniji se cepa.

  word-break: keep-all;

  word-break: break-all;

1. **Writing mode**

  writing-mode: horizontal-tb;

  writing-mode: vertical-rl;

**FONT**

- font family and how to use it : **<https://www.w3schools.com/css/css_font.asp>**

**- font-family :**

1. *****Serif*****
2. *****Sans-serif*****
3. *****Monospace:*****  have the same fixed width. They create a mechanical look.
4. *****Cursive:***** fonts imitate human handwriting.
5. *****Fantasy:***** fonts are decorative/playful fonts.

**Font fallback** - https://www.w3schools.com/css/css\_font\_fallbacks.asp

- it is very important to always use fallback fonts. This means that you should add a list of similar "backup fonts" in the font-family property. If the first font does not work, the browser will try the next one, and the next one, and so on. Always end the list with a generic font family name.

**Best web safe fonts:**

* Arial (sans-serif)
* Verdana (sans-serif)
* Tahoma (sans-serif)
* Trebuchet MS (sans-serif)
* Times New Roman (serif)
* Georgia (serif)
* Garamond (serif)
* Courier New (monospace)
* Brush Script MT (cursive)

**Font-style:**  normal, italic, oblique

**Weight:** light, medium, bold, black (debljina slova)

**Font-size:**  1em is 16px.

1vw = 1% of viewport width. If the viewport is 50cm wide, 1vw is 0.5cm.

Weight (zadebljanje fonta, normal, bold, lighter, bolder, ili u brojevima 100;)

**- Font parings :** https://www.w3schools.com/css/css\_font\_pairings.asp

**@font-face**

- koristi se kada npr ima kreiran specifican font za nekog klijenta pa ga mi hostujem na nas server

- na offline projektima

- bolja kontrola

**Fomati** : woff2, woff, ttf

 @font-face {

    font-family: 'Noto Sans KR';

    src: url(fonts/NotoSans/Sixtyfour\_Convergence/SixtyfourConvergence-Regular-VariableFont\_BLED\,SCAN\,XELA\,YELA.ttf),  format('ttf')

;

Nakon toga se u odredjenom elementu ili na body dodaje :

     font-family: 'Noto Sans KR';

**Eksterni** fontovi se moraju importovati :

    @import url('https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,100..900;1,100..900&family=Noto+Serif+Oriya:wght@400..700&display=swap');

- u elementu se pise ovo:

        font-family: "Montserrat", sans-serif;

**ICONS**

Ikone mozemo uzeti na Awesome Icon I Bootstrap !!!

**Awesome**: kreira se sopstveni **KIT**, to se kopira u **HEAD** a dole se dodaje icon koje mozemo naci na sajtu, najcesce su elementi i ili span.

Head : my kit za awesome:

   <script src="https://kit.fontawesome.com/6363efdda1.js" crossorigin="anonymous"></script>

Body :

   <i class="fa-solid fa-palette"></i>

Osnovni link za Awesome icons je :

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">

- nakon ovoga u HEAD-u, dole u BODY dodajemo ikonu koju zelimo.

Primer stilizovanja jedno pored drugog:

   <p style="display: inline;">Djordje ikona</p>

   <i class="fa-solid fa-palette" style="display: inline;"></i>

**BOOTSTRAP**

Head

   <link href="https://cdn.jsdelivr.net/npm/bootstrap-icons/font/bootstrap-icons.css" rel="stylesheet">

Body

   <i class="bi bi-pencil"></i>

**- starija verzija :**

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

**<i class="glyphicon glyphicon-cloud"></i>**

**<i class="glyphicon glyphicon-remove"></i>**

**<i class="glyphicon glyphicon-user"></i>**

**<i class="glyphicon glyphicon-envelope"></i>**

**<i class="glyphicon glyphicon-thumbs-up"></i>**

**….……………………………………………………………………………………**

**<p>Styled Bootstrap icons (size and color):</p>**

**<i class="glyphicon glyphicon-cloud" style="font-size:24px;"></i>**

**<i class="glyphicon glyphicon-cloud" style="font-size:36px;"></i>**

**<i class="glyphicon glyphicon-cloud" style="font-size:48px;color:red;"></i>**

**<i class="glyphicon glyphicon-cloud" style="font-size:60px;color:lightblue;"></i>**

**Google icons**

   <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

**LINKS**

: link (stil linka koji jos nije otvoren)

a: link {color: pink;}

: visited (stil linka koji je otvoren)

: hover (kada se stavi mis preko)

: active (kada se klikne na link)

: focus (Focus occurs when a browser discovers that you are ready to interact with an element on the page.)

- HOVER mora doci nakon visited i link a active mora nakon hover.

- Uz text-decoration se uklanja linija ispod linka.

a:link, a:visited {

  background-color: white;

  color: black;

  border: 2px solid green;

  padding: 10px 20px;

  text-align: center;

  text-decoration: none;

  display: inline-block;

}

a:hover, a:active {

  background-color: green;

  color: white;

}

**Cursor :**

    <span style="cursor:auto">auto</span><br>

    <span style="cursor:crosshair">crosshair</span><br>

    <span style="cursor:default">default</span><br>

    <span style="cursor:e-resize">e-resize</span><br>

    <span style="cursor:help">help</span><br>

    <span style="cursor:move">move</span><br>

    <span style="cursor:n-resize">n-resize</span><br>

    <span style="cursor:ne-resize">ne-resize</span><br>

    <span style="cursor:nw-resize">nw-resize</span><br>

    <span style="cursor:pointer">pointer</span><br>

    <span style="cursor:progress">progress</span><br>

    <span style="cursor:s-resize">s-resize</span><br>

    <span style="cursor:se-resize">se-resize</span><br>

    <span style="cursor:sw-resize">sw-resize</span><br>

    <span style="cursor:text">text</span><br>

    <span style="cursor:w-resize">w-resize</span><br>

    <span style="cursor:wait">wait</span><br>

**Display -** The display property is used to change the default display behavior of HTML elements.

* block (pocinje u novom redu I zauzima celu sirinu ekrana, moze se dodati visina I sirina).
* inline (ne moze mu se dodati visina I sirina, zauzima onoliko koliko zauzima sadrzaj).
* inline - block (formatiran je kao inline ali mu se moze dodati visina i sirina).
* none (sakriva taj element, ponasao se kao da ga ima ali se na stranici ne vidi). Isto se moze postici sa **Visibility : hidden.**
* list-item (elementi se ponasaju kao liste).

|  |  |
| --- | --- |
| **Value** | **Description** |
| inline | Displays an element as an inline element |
| block | Displays an element as a block element |
| contents | Makes the container disappear, making the child elements children of the element the next level up in the DOM |
| flex | Displays an element as a block-level flex container |
| grid | Displays an element as a block-level grid container |
| inline-block | Displays an element as an inline-level block container. The element itself is formatted as an inline element, but you can apply height and width values |
| inline-flex | Displays an element as an inline-level flex container |
| inline-grid | Displays an element as an inline-level grid container |
| inline-table | The element is displayed as an inline-level table |
| list-item | Let the element behave like a <li> element |
| run-in | Displays an element as either block or inline, depending on context |
| table | Let the element behave like a <table> element |
| table-caption | Let the element behave like a <caption> element |
| table-column-group | Let the element behave like a <colgroup> element |
| table-header-group | Let the element behave like a <thead> element |
| table-footer-group | Let the element behave like a <tfoot> element |
| table-row-group | Let the element behave like a <tbody> element |
| table-cell | Let the element behave like a <td> element |
| table-column | Let the element behave like a <col> element |
| table-row | Let the element behave like a <tr> element |
| none | The element is completely removed |
| initial | Sets this property to its default value |
| inherit | Inherits this property from its parent element |

The <script> element uses display: none; as default.

- ****Note:**** Setting the display property of an element only changes ****how the element is displayed****, NOT what kind of element it is. So, an inline element with display: block; is not allowed to have other block elements inside it.

visibility:hidden - Ne vidi se ali uzima prostor na stranici

display:none - Ne vidi se I ne uzima prostor na stranici

**Position**

* Static (normal flow of the page, standard position)
* Relative (adding **top, bottom, left, right** properties can cause to be adjusted away from normal position)
* Fixed (fixed, 4 other properties can be added. Positioned relative to the viewport)
* Absolute (positioned relative to the nearest positioned ancestor)
* Sticky (positioned based on the user's scroll position, You must specify at least one of top, right, bottom or left for sticky positioning to work)

1. **Index** - (ako je negativno, znaci da se pojavljuje iza drugog elementa, ako je pozitivno, onda je ispred drugog elementa. Ako ima vise elemenata preko kojih treba da se pojavljuje, stavlja se odredjeni broj (broj elemenata koje treba prekriti))

-**z-index**only works on **[positioned elements](https://www.w3schools.com/css/css_positioning.asp)**(position: absolute, position: relative, position: fixed, or position: sticky) and**[flex items](https://www.w3schools.com/css/css3_flexbox.asp)**(elements that are direct children of display: flex elements).

-If two positioned elements overlap each other without a z-index specified, the element defined ****last in the HTML code**** will be shown on top.

**Overflow**  - The overflow property only works for **block elements with a specified height**.

* Hidden (sakriva ono sto izlazi iz okvira elementa)
* Scroll (dodaje scrollbar)
* Auto (dodaje scrollbar gde god treba, horizontalno I vertikalno)
* Visible (Vidi se ono da visak izlazi iz elementa)
* Overflow-x (odredjuje kako se ponasa left and right)
* Overflow-y ( -II- top bottom)

  overflow-x: hidden; /\* Hide horizontal scrollbar \*/  
  overflow-y: scroll; /\* Add vertical scrollbar \*/

**Float** : left, right, none. (menja svojstvo block elementa I oni se pojavljuju jedan pored drugo ako se stavi left, right…moze biti I none I inherit).

- **CLEAR** - Da bi se neki element pojavi ispod, onda dodajemo ovaj property CLEAR

- left

- right

- none

- inherit

- both : znači: "Ovaj element ne sme stajati pored floatovanih elemenata ni sa leve ni sa desne strane."

**OVERFLOW : HIDDEN**

- U navigacionoj traci .topnav, koristi se float: left; za linkove (a). Kada se elementi koriste float, njihovi roditelji ne prepoznaju njihov visoki položaj. To može dovesti do problema sa layout-om, kao što je roditeljski element koji ima visinu 0 (ne prikazuje se pravilno). Kada se koristi overflow: hidden;, to "prisiljava" roditeljski element da uzme u obzir visinu svojih floatovanih elemenata, čime se sprečava da navigaciona traka postane nevidljiva ili da se ponaša nepredvidivo.

- https://www.w3schools.com/css/tryit.asp?filename=trycss\_website\_layout\_navbar

**Clearfix hack** - dodaje se u modernim kodovima da bi resio problem flotovanih elemenata u kontejneru. Ako neki od njih prevazilazi dimenzije roditelja, onda se dodaje ovaj clearfix kod I on ce to resiti.

.clearfix::after {

  content: "";

  clear: both;

  display: table;

}

- Kada koristiš float, može doći do situacije gde roditeljski element ne prepoznaje visinu svojih floatovanih elemenata. Tada se koristi clearfix:

* ::after dodaje nevidljivi element na kraj sadržaja.
* clear: both; kaže tom nevidljivom elementu da mora biti ispod svih floatovanih elemenata.
* display: table; osigurava da se taj nevidljivi element ponaša kao block element, tj. uzima prostor i "proširuje" roditeljski element tako da prepozna visinu svih svojih floatovanih elemenata.

**Zato** clear: both; **rešava problem floatovanja i omogućava roditeljskom elementu da zadrži ispravnu visinu.**

Compared to display: inline, the major difference is that display: inline-block allows to set a width and height on the element.

Also, with display: inline-block, the top and bottom margins/paddings are respected, but with display: inline they are not.

Compared to display: block, the major difference is that display: inline-block does not add a line-break after the element, so the element can sit next to other elements.

**WIDTH** - npr : staviti na 25% ako imamo 4 elementa u jednom, tako osiguravamo da ce sva 4 biti iste velicine. !!!

**Pseudo-classes:**

For example, it can be used to:

* Style an element when a user moves the mouse over it
* Style visited and unvisited links differently
* Style an element when it gets focus
* Style valid/invalid/required/optional form elements

**:focus -** (Moze se staviti kod formi. Kada se klikne na neki input element, da se fokusira, odnosno, na primer, promenimo boju pozadine). Mozemo izbrisati liniju koja se pojavljuje na borderu kada se klikne na elementa dodavanjem **outline: none**

**- ToolTip hover** : (div element, a nakon njega p sa ovim stajlom).

<style>

p {

    display: none;

    background-color: yellow;

    padding: 20px;

  }

  div:hover p { - div na hover izbacuje p element

    display: block;

  }

</style>

</head>

<body>

  <div>Hover over this div element to show the p element

    <p>Tada! Here I am!</p>

  </div>

</body>

**- First-child** (samo prvu p decu nekog elementa ili body)

p:first-child {

  color: blue;

}

**- Last child** (samo zadnji element u svim p elementima)

p i:last-child {

  color: blue;

}

**- First-child** (samo prvi p element I sve i elemente u njemu)

p:first-child i {

  color: blue;

}

**Pseudo-elements:**

For example, it can be used to:

* Style the first letter or line, of an element
* Insert content before or after an element
* Style the markers of list items
* Style the viewbox behind a dialog box
* Firste-line (da se doda poseban style na prvu liniju teksta) - p::first-line {} - only on **inline-block** elements

p:hover::first-line {

  color: #ff0000;

  font-variant: small-caps;

}

* Firste-letter (za style prvog slova) ::firste-letter. Only for **inline-block**. Moze se kombinovati sa html klasama.

p::first-letter {

  color: #ff0000;

  font-size: xx-large;

}

* Before (sluzi da se ubaci neki content pre kontenta nekog elementa) ::before

h1::before {

  content: url(smiley.gif);

}

* After (da se ubaci content nakog nekog kontenta) ::after

h1::after{

  content: url(smiley.gif);

}

* Selection (markira delove elementa koje je selektovao korisnik (color, background, cursor, outline) ) ::selection *(moze na sve ili na odredjeni element da se odnosi)*

::selection {

  color: red;

  background: yellow;

}

**DropDown**

<style>

.dropbtn{

background-color: #4CAF50;

color: white;

padding: 16px;

font-size: 16px;

border: none;

cursor: pointer;

}

.dropdown{

    position: relative;

    display: inline-block;

}

.dropdown-content{

    display: none;

    position: absolute;

    background-color: #f9f9f9;

    min-width: 160px;

    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);

    z-index: 1;

}

.dropdown-content a {

    color: black;

    padding: 12px 16px;

    text-decoration: none;

    display: block;

}

.dropdown-content a:hover {

    background-color: #f1f1f1;

}

.dropdown:hover .dropdown-content{   **/\* TOOLTIP HOVER \*/**

    display: block;

}

.dropdown:hover .dropbtn{

    background-color:  #3e8e41;

}

</style>

<link rel="stylesheet" href="work2.css">     <!-- external CSS -->

</head>

<body>

<div class="dropdown">

    <button class="dropbtn">Dropdown</button>

    <div class="dropdown-content">

        <a href="#">Link 1</a>

        <a href="#">Link 2</a>

        <a href="#">Link 3</a>

</div>

</div>

<p><strong>Note:</strong> We use href="#" for test links. In a real web site this would be URLs.</p>

</body>

**Dropdown picture**

<style>

.dropdown{

    position: relative;

    display: inline-block;

}

.content {

    display: none;

    position: absolute;

    background-color: #f9f9f9;

    min-width: 160px;

    z-index: 1;

    box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);

}

.dropdown:hover .content{

    display: block;

}

.desc{

    text-align: center;

    padding: 15px;

}

</style>

<link rel="stylesheet" href="work2.css">     <!-- external CSS -->

</head>

<body>

<div class="dropdown">

    <img src="icon.png" alt="error" width="100" height="50">

    <div class="content">

        <img src="anthurium-sch.jpg" alt="error" width="300" height="200">

        <div class="desc">Beautiful Cinque Terre</div>

    </div>

</div>

</body>

**Padajuci meni :**

- https://www.w3schools.com/css/tryit.asp?filename=trycss\_dropdown\_navbar2

## **Image Sprites**

- koristi se da se ubaci jedna slika koja sadrzi vise slika ili ikona u sebi. Na taj nacin stedimo prostor I smanjujemo broj konekcija na server za ucitavanje vise slika.

- Ovde zapravo se sece jedna slika na delove. Kada pise -40px 0px, to znaci da se sa leve strane odseca 40px I da se prikazuje ono sto sledi nakon toga. Ili ako je 0px -30px, to znaci da se sa leve strane ne odseca nista, ali se odozgo odseca 30px I prikazuje ono sto je ispod.

<head>

    <style>

    #navlist {

      position: relative;

    }

    #navlist li {

      margin: 0;

      padding: 0;

      list-style: none;

      position: absolute;

      top: 0;

    }

    #navlist li, #navlist a {

      height: 44px;

      display: block;

    }

    #home {

      left: 0px;

      width: 46px;

      background: url('img\_navsprites\_hover.gif') 0 0;

    }

    #prev {

      left: 63px;

      width: 43px;

      background: url('img\_navsprites\_hover.gif') -47px 0;

    }

    #next {

      left: 129px;

      width: 43px;

      background: url('img\_navsprites\_hover.gif') -91px 0;

    }

    #home a:hover {

      background: url('img\_navsprites\_hover.gif') 0 -45px;

    }

    #prev a:hover {

      background: url('img\_navsprites\_hover.gif') -47px -45px;

    }

    #next a:hover {

      background: url('img\_navsprites\_hover.gif') -91px -45px;

    }

    </style>

    </head>

    <body>

    <ul id="navlist">

      <li id="home"><a href="default.asp"></a></li>

      <li id="prev"><a href="css\_intro.asp"></a></li>

      <li id="next"><a href="css\_syntax.asp"></a></li>

    </ul>

    </body>

# **Attribute Selectors**

- **a[target="\_blank"]** { - menja css elementu sa atributom target I sa ovom vrednoscu !!!  
  background-color: yellow;  
}

**- a[target]** - samo sa ovim atributom

**- [title~="flower"]** - ima ovaj atribut a vrednost sadrzi rec flower (jedna od vise reci)

**- [class|="top"]** - oznacava element sa atributom class a vrednost je top ili sadrzi crticu -

**- [class^="top"]** - atribut class pocinje sa recju top

**- [class$="test"]** - zavrsava se sa recju test

**- [class\*="te"]** - selektuje element sa atributom class cija vrednost sadrzi te

**Counters:**

* counter-reset - Creates or resets a counter
* counter-increment - Increments a counter value
* content - Inserts generated content
* counter() or counters() function - Adds the value of a counter to an element

<style>

body {

  counter-reset: section;

}

h1 {

  counter-reset: subsection;

}

h1::before {

  counter-increment: section;

  content: "Section " counter(section) ". ";

}

h2::before {

  counter-increment: subsection;

  content: counter(section) "." counter(subsection) " ";

}

</style>

<h1>HTML/CSS Tutorials</h1>

<h2>HTML</h2>

<h2>CSS</h2>

<h2>Bootstrap</h2>

<h2>W3.CSS</h2>

<h1>Scripting Tutorials</h1>

<h2>JavaScript</h2>

<h2>jQuery</h2>

<h2>React</h2>

<h1>Programming Tutorials</h1>

<h2>Python</h2>

<h2>Java</h2>

<h2>C++</h2>

**Counters za liste :**

<style>

ol {

    counter-reset: section;

    list-style-type: none;

}

li{

    counter-increment: section;

    content: counters(section, ".") " ";

}

</style>

<ol>

    <li>item</li>

    <li>item

    <ol>

      <li>item</li>

      <li>item</li>

      <li>item

      <ol>

        <li>item</li>

        <li>item</li>

        <li>item</li>

      </ol>

      </li>

      <li>item</li>

    </ol>

    </li>

    <li>item</li>

    <li>item</li>

  </ol>

  <ol>

    <li>item</li>

    <li>item</li>

  </ol>

**UNITS:**

- <https://www.w3schools.com/css/css_units.asp>

**Math fonctions:**

calc() -   width: calc(100% - 100px); velicina je 100px manja od celog ekrana

 max()  -   width: max(50%, 300px); 50% ili 300px, sta je vece to postavlja

min() -   width: min(50%, 300px); 50% ili 300px, sta je manje to postavlja

**Gradient :**

- Koristi se za prelaz dve ili vise boja na pozadini: Moze biti:

* ****Linear Gradients (goes down/up/left/right/diagonally).** *Top-bottom default***

#grad1 {

  height: 200px;

  background-color: red; /\* For browsers that do not support gradients \*/

  background-image: linear-gradient(to right, red , yellow);

}

**- Dijagonalno :**

#grad {

  background-image: linear-gradient(to bottom right, red, yellow);

}

**- vise boja sa precentima koliko zauzimaju.**

#grad3 {

  height: 200px;

  background-image: linear-gradient(red 10%, green 85%, blue 90%);

}

**- transparency gradient**

#grad {

  background-image: linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1));

}

**- repeating gradient** : <https://www.w3schools.com/css/tryit.asp?filename=trycss3_gradient-linear_repeating>

* ****Radial Gradients (defined by their center)****

**- Default shape is ellipse, that can be changed.**

#grad {

  background-image: radial-gradient(circle, red 5%, yellow 15%, green 60%);

}

* ****Conic Gradients (rotated around a center point)****

**- Vise boja moze, moze im se odrediti koliko stepeni koja boja zauzima,**

#grad1 {

  height: 200px;

  width: 200px;

  background-color: red; /\* For browsers that do not support gradients \*/

  background-image: conic-gradient(red 45deg, yellow 90deg, green 210deg);

}

**- moze se dodati : border-radius: 50% da bi to bilo krug.**

**- repeating and postion of gradient : https://www.w3schools.com/css/css3\_gradients\_conic.asp**

**2D transforms**

div {

  transform: translate(50px, 100px);

}

**Transform :**

**- translate()**  - pomera element u odnosu na originalnu poziciju

**- rotate()** - transform: rotate(20deg); - moze biti I -20deg.

**- scaleX()** - horizontalno uvecava (width)

**- scaleY()** - vertikalno uvecava (height) -  transform: scaleY(0.5) - umanjuje za pola

**- scale()** -   transform: scale(2, 3); - uvecava 2x horizontalno I 3 puta vertikalno

**- skewX()** - horizontalno iskrivljuje po x osi, horizontalno

**- skewY()** - vertikalnno uskrivljuje po y osi, vetikalno

**- skew()** -   transform: skew(20deg, 10deg);

**- matrix()** - matrix(scaleX(), skewY(), skewX(), scaleY(), translateX(), translateY())

## **3D Transforms**

* rotateX() - rotira se po horizontalnoj osi
* rotateY() - rotira se po vertikalnoj osi
* rotateZ() - rotira se po Z osi po odredjenom stepenu   transform: rotateZ(90deg)

# **Transitions**

* transition
* transition-delay
* transition-duration
* transition-property
* transition-timing-function

 div {

  width: 100px;

  height: 100px;

  background: red;

  transition: width 2s, height 4s;

}

div:hover {

  width: 300px;

  height: 300px;

}

- Moramo odrediti I velicinu koja ce biti kada se predje misem preko elementa.

**Skracenica :**

div {  
  transition: width 2s linear 1s;  
}

**Transition-timing-function :**

* ease - specifies a transition effect with a slow start, then fast, then end slowly (this is default)
* linear - specifies a transition effect with the same speed from start to end
* ease-in - specifies a transition effect with a slow start
* ease-out - specifies a transition effect with a slow end
* ease-in-out - specifies a transition effect with a slow start and end
* cubic-bezier(n,n,n,n) - lets you define your own values in a cubic-bezier function

**Moze da se doda transform 2D ili 3D !!!!!!!!!!**

div {

  width: 100px;

  height: 100px;

  background: red;

  transition: width 2s, height 2s, transform 2s;

}

div:hover {

  width: 300px;

  height: 300px;

  transform: rotate(180deg);

}

**Skracenica**

div {

    width: 100px;

    height: 100px;

    background: red;

    transition: width 2s linear 1s, height 2s linear 1s, transform 2s linear 1s;

}

div:hover {

    width: 300px;

    height: 150px;

    transform: rotate(180deg);

}

Kroz transition možete menjati razne CSS osobine koje mogu imati glatku animaciju, uključujući:

1. **Dimenzije i položaj**: width, height, margin, padding, top, left, right, bottom
2. **Boje**: background-color, color, border-color, box-shadow
3. **Transfromacije**: transform (kao što su rotate, scale, translate, i skew)
4. **Prozirnost**: opacity
5. **Oblik ivica**: border-radius
6. **Vidljivost ivica**: border-width, border-style

Ne mogu se sve osobine animirati kroz transition, kao npr. display. Samo osobine koje mogu imati kontinualne vrednosti (brojevi, boje, i transformacije) su pogodne za transition.

**FLEX –** jednodimenzionalni prikaz

* **1. flex-direction** : (row, row-reverse : column, column-reverse) )
* flex-wrap : wrap, nowrap. Ako su zbir velicina item-a veca od glavnog diva, a stavimo wrap, onda ce se njihova velicina ispostovati a ostatak elemenata ce se prikazati u drugom recu inline. Ako stavimo nowrap, on ice se sabiti u jedan red da ispune velicinu glavnog diva. U kodu se moze pisati **flex-flow** (row wrap) (direction and wrap).
* Vrednost se moze ispisati skracenicom **flex-flow**
* Flex-grow (Koliko pozitivnog slobodnog prostora zauzima taj item. Kada mali divovi nemaju velicinu a glavni ima, onda se uz flex-grow ispunjava ceo glavni div. To se oznacava brojevima, 1,1,1,4,3. Broj oznacava koliko prostora ce zauzeti svaki od malih divova). Ako neki mali div ima velicinu, onda on se postavlja u toj velicini a ostali divovi se dele prema zadatoj proporciji.
* Flex-shrink (Koliko negativnog slobodnog prostora zauzima taj item) ?
* Flex-basis (Odredjuje velicinu item-a. moze biti auto, ili broj px). Tako se moze izdeliti glavni div na odredjene manje divove zadate velicine.
* Vrednost se ispisuje skracenicom u redosledu – grow, shrink, basis (flex: 2 2 10%;)
* Positive free space (kada je glavni div veci od divova u njemu, ostaje prazan prostor neispunjen)
* Negative free space (kada je glavni div manji od divova u njemu, mali divovi prelaze velicinu glavnog diva.)
* **2. Justify-content**: center, flex-start (default), flex-end, space-between. (horizontalno poravnjanje)
* **3. Align-items**: center, flex-start, flex-end, stretch (default), baseline (u sredini kao da je linija / <https://www.w3schools.com/css/tryit.asp?filename=trycss3_flexbox_align-items_baseline>) (vertikalno poravnjanje)
* **4. Align-content**: space-between, space-around, stretch (default), center, flex-start, flex-end.

**Flex items property**

* Order (<div class="flex-container">  
    <div style="order: 3">1</div>) </div>) (odredjuje u kom redosledu ce se pojaviti koji div).
* Flex-grow
* Flex-shrink
* Flex-basis
* Justify-self
* **Align-self** (odredjuje poravnjanje child elementa I preovladava u odnosu na align-item u glavnom elementu) center, flex-start, flex-end, stretch (default), baseline, auto - <https://www.w3schools.com/css/tryit.asp?filename=trycss3_flexbox_align-self_flex>

**Flex links :**

* https://stackoverflow.com/questions/32551291/in-css-flexbox-why-are-there-no-justify-items-and-justify-self-properties/33856609#33856609

**Flex Responsive**

* Koristi se da se odredi kako ce se sadrzaj pojavljivati na manjim ekranima, telefonima, tabletima…

@media (max-width: 800px) {

.flex-container {

flex-direction: column; }

Drugi nacin je da se u glavni element doda flex-wrap : wrap ; i da se u @media dodaju klase manjih divova kako bi se znalo da se na njih odnosi.

* <https://www.w3schools.com/css/css3_flexbox_responsive.asp>
* <https://www.w3schools.com/css/tryit.asp?filename=trycss3_flexbox_website2> / ceo sajt

**Transition (prvo je duration) -** <https://www.w3schools.com/css/css3_transitions.asp>

1. **transition-property**

background-color, background-position

border-color, border-width, border-spacing

left, rigth, top, bottom

color, opacity

font-size, font-weight, letter-spacing, word-spacing, line-height, text-indent

width, height, max-width, max-heaight, min-width, min-height

padding, margin

text-shadow

visibility

z-index

vertical-align

outline-color, outline-offset, outline-width

1. **transition-duration (npr 2s)**

* Koristi se da se izmeni tranzicija osnovnog stila. Koristi se :hover, sto oznacava da se tranzicija aplikuje kada se stavi mis na odredjeni element.
* <https://www.w3schools.com/css/css3_transitions.asp>

1. **div {transition-timing-fonction}** - Odredjuje brzinu efekta koji mogu biti

* ease - specifies a transition effect with a slow start, then fast, then end slowly (this is default)
* linear - specifies a transition effect with the same speed from start to end
* ease-in - specifies a transition effect with a slow start
* ease-out - specifies a transition effect with a slow end
* ease-in-out - specifies a transition effect with a slow start and end
* cubic-bezier(n,n,n,n) - lets you define your own values in a cubic-bezier function

1. **div {transition-delay}  -** odredjuje vreme nakon koga ce se aplikovati tranzicija. Odredjuje se u sekundama 1s.
2. **div {transform}** odredjuje kako ce se element transformisati (rotiranje npr). Potrebno je staviti u :hover koja transformacija ce biti uradjena a u transition se pise kojom brzinom.

* -webkit-transform: scale(1.5, 1.5);

**Transition vendor prefixes**

* -webkit- (chrome, safari, newer versions of opera)
* -moz- (firefox)
* -o- (old version of opera)
* -ms- (internet explorer, edge)
* -webkit-transition-duration: 1s; -

**Animation**

* 1. **animation-name**
* Prvo se odredjuje ime u **@keyframes** (@keyframes IME), a onda se u glavnim karakteristikama dodaju promene u snimaciji.

div

{

height: 100px;

background-color: red;

animation-name: example;

animation-duration: 4s;

}

@keyframes example {

from {background-color: red;}

to {background-color: yellow;}

}

* U sledecem primeru se vidi da se moze menjati vise boja :
  1. **animation-duration 4s ()**

div {

height: 100px;

background-color: red;

animation-name: example;

animation-duration: 4s;

}

@keyframes example {

0% {background-color: red;}

25% {background-color: yellow;}

50% {background-color: blue;}

100% {background-color: green;}

}

* U ovom primeru se vidi da se moze menjati i pozicija elementa

div {

height: 100px;

background-color: red;

position: relative;

animation-name: example;

animation-duration: 4s;

}

@keyframes example {

0% {background-color:red; left:0px; top:0px;}

25% {background-color:yellow; left:200px; top:0px;}

50% {background-color:blue; left:200px; top:200px;}

75% {background-color:green; left:0px; top:200px;}

100% {background-color:red; left:0px; top:0px;}

}

**3. animation-delay** (vreme za koje ce poceti izvrsenje animacije, mogu se pisati obicni brojevi ili negativni brojevi sto znaci da ce izgledati kao da se vec odvija animacija.)

**4. animation-iteration-count** (podesava koliko puta ce se izvrsiti animacija. U brojevima il infinite)

**5. animation-direction** (kako se pojavljuje. Moze biti : normal, reverse, alternate (brze), alternate-reverse).

**6. animation-timing-function** (isto kao kod tranzicije)

**7. animation-fill-mode** (prikacuje kako izgleda element kada animacija nije igrana)

***Zapazanja***

1. ***Odraditi : prezentacija 20 I cas 11. RESPONSIVE (prezentacija 16 da se negde nauci)***
2. **Line-height** – odvaja jedan red teksta od drugog. Moze se koristiti za pozicioniranje teksta, vertikalno. (div height 200px ; p line-height200px = tekst na sredini).
3. **Clear** – odredjuje sta ce se desiti sa elementom pored plutajuceg elementa. Sto znaci da jedan mora biti FLOAT a drugom dodajemo CLEAR.