```
1. Codepen: can use for better structuring of HTML, CSS and JS

2. HTML: markup language(means descriptive language, set of tags)
   2.1 Tags(Elements): tags are keywords surrounded by <angle> brackets:
       Opening    Closing              Tags
       '<html>'   '</html>'
       '<head>'   '</head>'            Meta-data section
       '<body>'   '</body>'
       '<p>'      '</p>'
       '<h1>'     '</h1>'              From <h1> to <h1> - largest to smallest
       '<strong>' '/strong>'          All of this text will appear in bold<b>
       '<em>'     '</em>'             All of this text will appear in italic<i>
       '<sup>'    '</sup>'            Superscript
       '<sub>'    '</sub>'            Subscript
       '<aside>'  '</aside>'          If we want to add something additional to
the main content
       <a a href="#">Click here</a>   Anchor Links
       <img src="/images/logo.png">   Images
       <ol>         </ol>            Ordered list (e.g.:list-style-type:none)
       <ul>         </ul>            Unordered list
       <dl>                          Definition list
           <dt></dt>                 Definition title
           <dd></dd>                 Definition description
           <dt></dt>
           <dd></dd>
                    </dl>
       <hr>                          adds line in the page
       <br>                          adds empty row
       <details> <summary> <p>       semantic details and summary
       <blockquote>                  цитат

   2.2 Attributes: provide additional information about HTML.
       Tag elements can have Attributes
       - always use "double style quotes" for attributes;

   3.3 Additional Info:
       - Example of how to open a page in a new tab:
           <a href="new-file.html" target="_blank">New File</a>

3. CSS: Cascading Style Sheets
   3.1 Selectors: to which HTML elements our CSS will be applied
       [selector]{
           [declaration];
           [declaration];
           color: beige;
           [property: value];
       }

       3.1.1 Element/Type Selectors:
       h1{}

       3.1.2 Class Selectors:
       .class_name{}

       3.1.3 ID Selector:
       #id_name{}

       3.1.4 Attribute Selector:
       [href="https://softuni.bg"]{}
```

```
        a[href^="http"] -> all a's that start with http
        a[href$=".com"] -> all a's that end with .com
        a[href*="ubb"] -> all a's that contain ubb inside (e.g. https://ubb.com)

        3.1.5 Universal Selector(*) - selects all of the elements:
        *{}

        3.1.6 Descendant Combinator:
        The elements matched by the second selector are selected if they have an
        ancestor(parent, parent't parent) element matching the first selector.
        Select all the paragraphs that are in the header
        header p{}

        3.1.7 Selector List:
        Selects all the matching elements.
        Select all headers, p and div elemenets
        header, p, div{}

        3.1.8 Child Combinator:
        Matches only those elements matched by the second selector that are the
direct
        children of elements matched bu the first.
        header > p{}

        3.1.9 All of paragraphs in front of which there is a paragraph
        p + p {}

    3.2 3 Ways of Inserting a CSS:
        3.2.1 External style sheet
        <head>
            <link rel="stylesheet" type="text/css" href="styles.css">
        </head>

        3.2.2 Internal style sheet
        Used when a single document has a unique style. You define internal styles
in the
        head section of an HTML page, by using the <style> tag.
        <head>
            <style>
                body{
                    ...
                }

        3.2.3 Inline style
        Do not use this method unless you have no other choice
        < div style="color:red;">

        3.2.4 Using !important

    3.3 Inheritance:

    [0]                 [0]                 [0]                 [0]
    Inline              ID                  classes,            elements,
                                            attributes,         pseudo-elements
                                            pseudo-classes

    HIGHEST ------------------------------------------> LOWEST

    3.4 CSS Properties:
```

```
        - background
        - text (color; text-align; text-decoration; text-transform; text-indent)
                - text-align: left(да подравним текста от ляво)
                - text-decoration: underline
                - text-indent: 5em (дали искаме да преместим текста да е с индентация)
                - text-overflow: ellipsis; (като няма място да отреже текста и да сложи
...)
                - text-shadow: 0 0 5px #000 (да добавим падаща сянка)
        - font
        - border

    3.5 Additional Info:
        - display: inline-block; means all of the elemenets to be in the same
row(e.g.: nav)
        - display: table-cell; make the articles one to another

4. Typography:
    4.1 Styling text:
        - Measuse - the numbers of characters in a single line(between 45 and
75char. e.g.: width: 66ch;)
        - Leading("ledding") - the space between the lines(line-height - 1.5 is
good)
        - Font Family:
                - serif - със завъртулки
                - sans-serif - без завъртулки - печатно
                - monospace - всеки 1 символ заема еднакво място на страницата
                - cursive - ръкопис
                - fantasy - доста стилизиран текст
5. CSS Units:
    - px
    - em
    - rem - reference to the main font size

6. Google Fonts:
    - In CSS File on top import:
    @import url('https://fonts.googleapis.com/css2?
family=Roboto:ital,wght@0,100;1,100&display=swap');
    -Then add to font-family
    html{
        font-family: 'Roboto', sans-serif;
    }

7. Font Awesome:
    - In the CSS files:
    @import url(https://use.fontawesome.com/releases/v6.4.0/css/all.css);
    @import url("https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.2/css/
all.min.css")
    - In the HTML file:
    <button class="button fill"><i class="fa-solid fa-arrow-down"></i>
Download</button>

.RESET.CSS:
*{
    padding: 0;
    margin: 0;
}
.TYPOGRAPHY.CSS:
body, table, input, button{
    font: inherit;
```

```css
}

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

html{
    font: 16px/1.5 'Times New Roman';
        box-sizing: border-box;
}
```

----
1. Display Box Model:
    1.1. Block Box - take the whole space of the row, create 1empty row before and
after themselves (=layout elements)
    1.2. Inline Box - part of the text (e.g.: span, a)
        - overflow: scroll, hidden, visible
    1.3. Inline Block Box - gives the ability to use vertical padding and margin on
inline elements as well as adding width and height

2. The Standard Box Model:
        - box-sizing: content-box; <- default one
        - box-sizing: border-box;

3. Position:
        - static(default)
        - relative(move it towards the main position of the element in the layout)
        - absolute(positions the element towards the closest element with non-
static position)
        - fixed(positions elements towards the browers' viewport)
    3.1 Z-index: largest index is on top (goes well with position relative)

4. FLEX CONTAINER - box is flex container:
    .box {
        display: flex;
        flex-direction: row, column, row-reverse, column;
        /* by default is nowrap */
        flex-wrap: wrap;
        /* align by horizontal */
        justify-content: flex-start, center, flex-end, space-around, space-between,
space-evenly;
        /* align by vertical, stretch by default */
        align-items: stretch, flex-end, center, flex-start;
        /* gap: up, down, left, right */
        gap: 2em;
    }
    align items vs align content

5. FLEX ITEMS - items properties:
    .box{
        order: 1;
        /* initial - fixed as initial; 1-item can grow, ALSO can use for
proportions - GOES WELL WITH flex-basis: auto;*/
        flex-grow: 1, initial;
        flex-shrink: 1;
        /* flex-basis == basic width */
        flex-basis: 20em;
```

```
    }

<----------> main axis-justify-content(by horizontal ) NOTE: when NO REVERSE => if
reverse it is the opposite
^
|
|
|
cross axis-align-items(by vertical)


--------------------------------------------------------------------------------
---------------------------
1. Media Queries:

    @media only screen and (max-width: 400px//orientation: landscape, hover:hover,
aspect-ratio: 1/1){
        ...
    }

    @media all ...

    @media screen, print{
        ...
    }

    @import url ('styles.css') screen;

    CALC 16 / 9 -> (9/16) * 100 for padding-top
    inset 0 == top 0 bottom 0 left 0 right 0
    appearance: none -> thing not to be visualized
    pointer-events: none -> e.g: checkbox cannot be checked

2. Container Queries:
    @container sidebar (min-width: 700px) {
        .card{
            font-size: 2em;
        }
    }

    circle{
        width: 4em;
        height: 4em;
        border-radius: 50%;
        content: '';
        display: block;
        background: green;
    }

3. Forms:
    3.1. Input Types:
    - text input
    - textarea
    - radio buttons
    - dropdown menu
    - button

    <FORM METHOD='GET/POST' ACTION='/SERVER/'>
```

3.1. Label: the HTML label element represents a caption for an item in a user
interface
        <Label FOR='FULL-NAME'> NAME</LABEL>
        <INPUT ID='FULL-NAME' NAME='FULL-NAME' TYPE='TEXT'/>

--- HOT KEYS &
URLs---

```
OPENING TAG LETTER/WORD + TAB    - creates both opening and closing tags
ALT + SHIFT + DOWN ARROW         - copies the current row below
HOME                             - go at the beginning of the row
END                              - go at the end of the row
F12                              - inspect code
SHIFT + ALT + F                  - formatting code
&copy;                           - crates the "R" for all rights reserved
p*3>a*5                          - creates 3paragraphs with 5 a's in each paragraph
```

- TO GOOGLE SEARCH USE: 'mdn...'=Mozilla Developement Network e.g.:'mdn list
types'
- TO EXPLAIN SELECTORS USE: https://kittygiraudel.github.io/selectors-explained/?
ref=tiny-helpers&s=article%2520%7E%2520p
- @import from Google Fonts if we want to load a new different font Family
- TO FIND COLORS THAT COMBINE WELL: UI Color Platte(HINT: use 'opacity' for colours
- прозрачност)
- caniuse.com - check if selectors can be used for different browsers
- W3C Markup Validation Service - check whether your HTML is written correctly
- Gradient linear generator (can use for background color)
- For image in text can use:
```
    img{
        display: block;
        align: right;
    }

    .title{
        space-between: 1ch;
    }
```
- For box shadow can use:
```
    box-shadow: 0 0 1em 0 rgba(0,0,0,0.2);
```
-For centering the content of the site can use:
```
    margin: 0 auto; --> centers the content of the site
```
- For list style can use:
```
    list-style-position: inside;
```