**html\_css\_exam**

**FontAwsomeIcons:**

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v**6.1.1**/css/all.css> **5.15.4**

@import url("https://use.fontawesome.com/releases/v5.15.4/css/all.css");

**overflow:** auto за скрол на счупени елементи hidden за без скрол scroll за двоен скрол

**Text-decoration**

Underline

**List-style**

Square

cursor: pointer;

**Оразмеряване**

box-sizing: border-box така расте навътре при увеличаване на padding и border. При content-box расте обратно навън и размества всичко!

**Центриране**

text align – center центрира съдържание на елемент

**margin: 0 auto** центрира блоков елемент по хоризонтала

Background image – за уеднаквяване на размери на снимки и ограничаането им, за да не счупят страницата. Подобно става с object-fit:cover !

img{

width: 100%;

height: 100%;

object-fit: cover;

}

Aside и main се задават като display:inline или ib, за да се позиционират сами!

Floats

Клас на съдържащия контейнер

.clearfix::after {

content: "";

clear: both;

display: table;

}

linear gradient – за преливка на цвят:

    background-image: linear-gradient(to right, #a83278, #d28213);

пишеш color:red в браузъра и взимаш цвета с пипетата...

position: absolute;

    top: 50%;

    left: 50%;

    transform: translate(-50%, -50%);

...за центриране

.card{

    margin: auto;

    width: 50%;

}

… центриране на card.

**GRID simple**

ul {

display: grid;

grid-template-columns: 100px repeat(2, 1fr) 200px;

*grid-template-columns: repeat(auto-fill, 200px); за брутално наместване на колони по размера на екрана!*

grid-template-rows: 300px;

gap: 12px 20px;

}

**GRID templating**

body {

display: grid;

grid-template-columns: repeat(4, 1fr);

grid-template:

“head head head head”

“aside main main main”

“aside . . . ”

“foot foot foot foot”

grid-auto-columns: minmax(0px, 1fr);

grid-auto-rows: minmax(0px , 1fr);

gap: 12px;

}

.header {

grid-area: head;

}

.aside {

grid-area: aside;

}

Lazy loading:

Img {

loading=”lazy”;

}

**ПРМЕНЛИВИ**

:root {

--clr-black: #191919;

--clr-white: #ffffff;

}

.btn {

var(--cvlr-black);

background-color(--clr-white);

}

**Псевдо** CSS чилдрени:

.section::before { .section::after {

content: “”; content: “”;

} }

**Transition & hover**

.nav-item .link {

color: white;

transition: color 300ms ease-in-out;

}

.nav-item .link:hover{

color: red;

}

**BackgroundImg**

.imgHolder {

height: 106px;

background-image: url(…);

backgroung-position: center;

background-size: cover *реже ръбовете, когато излиза пропорцията, но единия размер го прави по рамката!*

}

**Not селектор**

#about p:not(:last-child){

margin-botton: 1rem;

}

*Ще важи за всички р без last child!*

img {

**transform: scale(1.5);**

**}** *ще увеличи снимката с 50%*

**Full HD 1920 x 1080**

**Background Opacity**

#header::after{

content: ‘ ‘;

position: absolute;

top: 0px;

left: 0px;

width: 100%;

height: 100%

background-color: black;

z-index: -1;

opacity: 0.5;

**Background**

header img.bg{

    position: absolute;

    top: 0px;

    left: 0px;

    width: 100%;

    height: 100%;

    object-fit: cover;

    z-index: -1;

}

<https://github.com/simeon-petrov-5/SoftUni-HTML-CSS-May-2022>

CSS SELECTOR COMBINATORS - <https://developer.mozilla.org/en-US/docs/Web/CSS/Child_combinator> | <https://www.youtube.com/watch?v=cuEHx9DoWI4>

<https://www.w3schools.com/css/css_specificity.asp> | <https://css-tricks.com/specifics-on-css-specificity/>

## Lecture 4 CSS Box model

* [Generating font-size CSS Rules and Creating a Fluid Type Scale](https://moderncss.dev/generating-font-size-css-rules-and-creating-a-fluid-type-scale/)

## Esercise 4 CSS Box model

* [meyerweb reset](https://meyerweb.com/eric/tools/css/reset/)
* [A Modern CSS Reset](https://piccalil.li/blog/a-modern-css-reset/)
* [Fighting the Space Between Inline Block Elements](https://css-tricks.com/fighting-the-space-between-inline-block-elements/)
* [How to remove the space between inline/inline-block elements?](https://stackoverflow.com/questions/5078239/how-to-remove-the-space-between-inline-inline-block-elements)

## CSS Positioning & GRID

* [Learn Z-Index Using a Visualization Tool](https://thirumanikandan.com/posts/learn-z-index-using-a-visualization-tool)

## Esercise CSS Positioning & GRID

* [caniuse](https://caniuse.com/)
* [complete-guide-grid](https://css-tricks.com/snippets/css/complete-guide-grid/)
* [Figma - Grid layout design](https://www.figma.com/community/file/978648046298721882)

<https://stackoverflow.com/questions/44488357/equal-height-rows-in-css-grid-layout> На 1вия отговор има едно болднато EXCEPT. В него пише, че ако размерите на parent-a са неопределени, както е в случая, grid-auto-rows: 1fr няма да работи

<https://www.google.com/search?q=peter+griffin+css&sxsrf=ALiCzsYT95mqE30pyx0eTiv_UKysLz4yqw:1654801301896&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjivpeeh6H4AhXJQvEDHR6_AswQ_AUoAXoECAEQAw&biw=1536&bih=758&dpr=1.25#imgrc=X6Dvnt573OqzrM>