



Omówienie trzeciej pracy domowej

# Selektory

```
h2 {  
  padding: 40px;  
  margin-top: 40px;  
  font-family: OpenSans, Helvetica;  
  color: #EF008E;  
  font-size: 28px;  
  letter-spacing: 3.8px;  
  font-weight: bold;  
  text-transform: uppercase;  
}  
  
.page-navigation a {  
  font-family: OpenSans, Helvetica;  
  color: #EF008E;  
  font-size: 13px;  
  letter-spacing: 6.7px;  
  font-weight: bold;  
  text-transform: uppercase;  
  text-decoration: none;  
}
```

```
#video {  
  display: flex;  
  position: relative;  
  justify-items: center;  
  align-items: center;  
}
```

Nie stylujemy po:

- tagach
- id

Nie piszemy selektorów takich jak .klasa tag, raczej też .klasa .inna-klasa.

# Font-family w wielu miejscach

```
.content-section {  
  min-height: 100vh;  
  font-family: OpenSans, Helvetica;  
  font-style: normal;  
  color: #204280;  
  font-size: 13px;  
}  
  
h2 {  
  padding: 40px;  
  margin-top: 40px;  
  font-family: OpenSans, Helvetica;  
  color: #EF008E;  
  font-size: 28px;  
  letter-spacing: 3.8px;  
  font-weight: bold;  
  text-transform: uppercase;  
}  
  
.page-navigation a {  
  font-family: OpenSans, Helvetica;  
  color: #EF008E;  
  font-size: 13px;  
  letter-spacing: 6.7px;  
  font-weight: bold;  
  text-transform: uppercase;  
  text-decoration: none;  
}
```

Trzy razy definiowane takie samo font-family. Wystarczy raz, wyżej, na przykład na body. Jeżeli gdzieś musi być inna czcionka - wystarczy wtedy nadpisać.

# Letter spacing

```
letter-spacing: 3.8px;  
font-weight: bold;  
text-transform: uppercase;
```

Letter-spacing był dość często pomijany przy stylowaniu tekstu. To detal, na który jednak warto zwracać uwagę.

# Wartości reguł

```
width: 10 px;  
}
```

```
.form__label {  
  color: dbd8fb;  
  text-align: left;  
  color: #ccc8ff;  
  opacity: 50%;  
  letter-spacing: 1.5px;  
  display: flex;
```

Zdarzały się niewłaściwe wartości, takie z błędem (10 px - ze spacją), bądź w złych jednostkach (opacity: 50% - opacity przyjmuje wartości od 0 do 1, gdzie 1 to 100%). Prawidłowe wartości, takie jakie dana reguła przyjmuje, możecie sprawdzić na [Mozilla Developer Network](https://developer.mozilla.org/pl/docs/Web/CSS), bądź w omawianych na poprzednich zajęciach devToolsach - po zaznaczeniu danego elementu pojawią nam się jego style, jeśli któryś z nich będzie nieprawidłowy - pokaże nam się wykrzyknik.

# Float

```
float: right;  
margin-right: 64px;  
display: inline-block;
```

Float nie służy do pozycjonowania elementów na stronie - znacznie lepiej używać flexa bądź innych sposobów (position relative/absolute, display inline-block, flex, ...). Float zachowuje się nieintuicyjnie i wymaga dziwnych tricków - a nie chcemy utrudniać sobie życia :)

Przy okazji - jeżeli ustawimy regułę float na elemencie, to display zawsze przyjmie wartość "block", niezależnie od tego co wpisujemy.

## Universal selector - \*

```
* {  
  padding: 0 0 0 0;  
  margin: 0 0 0 0;  
  background-attachment: fixed;  
  background-image: url(img/background.svg);  
}
```

Zdecydowanie nie warto go nadużywać - przydaje się najczęściej do ustawienia wartości box-sizing. W przypadku po lewej każdy element na stronie, który nie nadpisuje wartości background-image będzie miał tło background.svg.

# Style inline

```
<div style="display: flex; flex-wrap: wrap; justify-content: space-between">
  <div style='height: 20vh; width: 25%; text-align: center; display: flex; flex-wrap: w
    <div class="bolded" style="letter-spacing: 4.5px; font-size: 15px; text-align: left";
  </div>
  <div style='height: 20vh; width: 25%; text-align: center; display: flex; flex-wrap: w
    <div class="bolded" style="letter-spacing: 4.5px; font-size: 15px; text-align: left";
  </div>
  <div style='height: 20vh; width: 25%; text-align: center; display: flex; flex-wrap: w
    <div class="bolded" style="letter-spacing: 4.5px; font-size: 15px; text-align: left";
  </div>
  <div style='height: 20vh; width: 25%; text-align: center; display: flex; flex-wrap: w
    <div class="bolded" style="letter-spacing: 4.5px; font-size: 15px; text-align: left";
  </div>
```

Style inline znacznie zmniejszają czytelność kodu HTML i utrudniają stylowanie w CSSach - bo nadpiszą każdą regułę którą napisalibyśmy w pliku CSS. Ponadto dobrą praktyką jest rozdzielanie warstwy treści (HTML) od warstwy prezentacji (CSS).



# Wcięcia i formatowanie kodu

```
body{
background-image: url("background.s
background-size: cover;
background-attachment: fixed;
margin: 0px;
}

.menu{
color: #ec008c;
font-weight: bold;
font-size: 13px;
letter-spacing: 6.7px;
font-family: "Helvetica";
position: top;
}
```

Odpowiednie formatowanie kodu znacznie zwiększa jego czytelność i jest pierwszą rzeczą którą widzi czytający. Oczywiście nie musicie przeglądać całego kodu samemu - są do tego odpowiednie wtyczki w edytorach. VSCode posiada wbudowaną funkcję “Format Document” ukrytą domyślnie pod kombinacją przycisków Shift + Alt + F (domyślnie, ewentualnie można ją wyszukać w Google).



W razie wątpliwości - pytajcie!