

# Inne elementy CSS

1. Listy
2. Tabele
3. Pozycjonowanie
4. Sposoby wyświetlania elementów

# 1 Listy

## Styl listy - `list-style-type`

- selektor `{list-style-type: typ; }`
- Dotyczy: `<ul>`, `<ol>`, `<li>`.
- Typy punktatorów:
  - disc
  - circle
  - square
  - decimal
  - lower-roman
  - upper-roman
  - lower-alpha
  - upper-alpha
  - none

# 1 Listy

## Obraz jako punktork - `list-style-image`

- selektor `{list-style-image: url(ścieżka_dostępu);}`

# 1 Listy

## Pozycja punktora - `list-style-position`

- selektor `{list-style-position: pozycja;`
- Parametr *pozycja* określa zachowanie się tekstu, który nie mieści się w jednej linii, względem punktora. Możliwe są dwie wersje:
  - `outside`
  - `inside`

## 2 Tabele

### Podpis - `caption-side`

- selektor `{caption-side: ustawienie;`
- Parametr *ustawienie* może przyjmować wartości:
  - `top`
  - `bottom`
  - `left`
  - `right`

## 2 Tabele

### Obramowanie - border

- selektor {border: *wartości*;}  
• Dla tego atrybutu można ustawić grubość, rodzaj oraz kolor obramowania.
- **Np.** `table, tr, td {border: 2px solid red;}`

## 2 Tabele

### Sposób obramowania

- selektor `{border-collapse: model;`
- Wartości parametru *model* :
  - collapse
  - separate

### Sposób obramowania pustych komórek

- selektor `{empty-cells: wartość;`
- Wartości atrybutu:
  - show
  - hide

# 2 Tabele

## Rozmieszczenie - `table-layout`

- selektor `{table-layout:  
rozmieszczenie;`
- Wartości atrybutu:
  - auto
  - fixed



## 2 Tabele

### Odstępy między komórkami

- selektor `{border-spacing: odstęp;}`
- Odstęp w jednostkach długości.
  - Jedna wartość
  - Dwie wartości (definiujemy poziomy oraz pionowy odstęp między komórkami)

### Szerokość i wysokość tabeli

- Za pomocą właściwości `width` i `height`:

# 3 Pozycjonowanie

## position

- selektor `{position:rodzaj; parametry; }`
- Jako rodzaj:
  - static
  - relative
  - absolute
  - fixed
- Jako parametr (przesunie o określoną wartość w stosunku krawędzi):
  - left: wartość;
  - top: wartość;
  - right: wartość;
  - bottom: wartość;

# 3 Pozycjonowanie

## `position:static;`

- `static` – przywraca normalne pozycjonowanie elementu, czyli jeżeli wcześniej (np. w arkuszu stylów) została podana deklaracja pozycjonowania tego elementu
- selektor `{position:static; parametry;}`
- Jako parametr (przesunie o określoną wartość w stosunku krawędzi):
  - `left`: wartość;
  - `top`: wartość;
  - `right`: wartość;
  - `bottom`: wartość;

# 3 Pozycjonowanie

## `position: relative;`

- `relative` – (pozycjonowanie względne) pozwala przesunąć wybrany element w inne miejsce w stosunku do położenia pierwotnego
- selektor `{position: relative; parameters; }`
- Jako parametr (przesunie o określoną wartość w stosunku krawędzi):
  - `left`: wartość;
  - `top`: wartość;
  - `right`: wartość;
  - `bottom`: wartość;

# 3 Pozycjonowanie

## `position: absolute;`

- `absolute` – pozwala przesunąć wybrany element w inne miejsce względem brzegów strony lub bloku
- selektor `{position: absolute; parametry; }`
- Jako parametr (przesunie o określoną wartość w stosunku krawędzi):
  - `left`: wartość;
  - `top`: wartość;
  - `right`: wartość;
  - `bottom`: wartość;

# 3 Pozycjonowanie

## `position: fixed;`

- `fixed` – działa podobnie jak `absolute`, z tą różnicą, że ustala położenie elementu zawsze względem krawędzi okna przeglądarki. Tak pozycjonowany element jest nieruchomy przy przewijaniu strony (jest cały czas widoczny w tym samym miejscu)
- selektor `{position: fixed; parametry;}`
- Jako parametr (przesunie o określoną wartość w stosunku krawędzi):
  - `left`: wartość;
  - `top`: wartość;
  - `right`: wartość;
  - `bottom`: wartość;

## 3 Pozycjonowanie

Przezroczystość – `opacity` – pozwala zdefiniować przezroczystość dla elementu wstawianego na stronę

- selektor  
`{opacity: nieprzezroczystość; }`
- Parametr `nieprzezroczystość` to liczba z przedziału od 0,0 do 1,0. 0 oznacza całkowitą przezroczystość, 1 – brak przezroczystości.

# 3 Pozycjonowanie

**Nakładanie elementów (warstwy)** – gdy kilka elementów nachodzi na siebie, pozwala określić kolejność nakładanych elementów

- selektor `{position: rodzaj;  
parametry; z-index: numer;`
- Parametr `numer` mówi o kolejności nakładania pozycjonowanych elementów. Element z numerem 1 będzie zakryty przez inne elementy z wyższym numerem



### 3 Pozycjonowanie

**Obcinanie elementu** – do wycięcia z elementu w kształcie prostokąta służy atrybut

```
position: absolute;  
clip: rect(współrzędne w  
jednostkach długości)
```

- selektor {position: *absolute*;  
clip: rect(*góra*, *prawo*, *dół*,  
*lewo*); }
- Można używać określenia `auto` – co oznacza cięcia z danej strony

# 3 Pozycjonowanie

**Wyrównanie w pionie** – pozwala na zdefiniowanie wyrównania elementu w stosunku do innych elementów strony

- selektor `{vertical-align: sposób;`

Parametr *sposób*

- `baseline` – wyrównuje linię bazową do linii elementu nadrzędnego
- `middle` – ustawia element na środku wysokości elementów sąsiadujących
- `text-top` – wyrównuje do górnej krawędzi tekstu elementu nadrzędnego
- `text-bottom` – j.w. tylko do dolnej
- `super` – ustawia element jako indeks górny
- `sub` – ustawia element jako indeks górny
- `top` – wyrównuje do górnej krawędzi najwyższego elementu w linii
- `bottom` – wyrównuje do podstawy najniższego elementu w linii
- w jednostkach wartości

# 3 Pozycjonowanie

**Ustawienie w poziomie – „pływanie” po lewej lub prawej stronie okna przeglądarki lub bloku**

- selektor {float: *sposób*;}

Parametr *sposób*

- `left` – element zostanie ustawiony po lewej stronie względem sąsiednich elementów
- `right` – element zostanie ustawiony po prawej stronie względem sąsiednich elementów
- `none` – element nie będzie sąsiadował z innymi elementami

# 3 Pozycjonowanie

Jeżeli tekst umieszczony z prawej lub z lewej strony zdjęcia będzie zbyt krótki, to zdjęcia nie zostaną ułożone jedno pod drugim, lecz znajdą się obok siebie. Aby temu zapobiec należy użyć atrybutu `clear`, który ustala pozycję kolejnych elementów względem elementu pływającego

- selektor `{clear: sposób;}`

Parametr `sposób`

- `left` – lewy bok następnego elementu nie przylega do elementu pływającego
- `right` – prawy bok następnego elementu nie przylega do elementu pływającego
- `both` – żaden bok następnego elementu nie przylega do elementu pływającego
- `none` – brak ograniczeń

## 3 Pozycjonowanie

**Kolumny na stronie** – atrybut `float` może być wykorzystywany do tworzenia kolumn w dokumencie poprzez umieszczenia zawartości w `<div>`

- `div {float: sposób;  
cecha (np. (szerokość) :wartość}`

Np.: Po zdefiniowaniu dla każdego znacznika `<div>` atrybutu `float:left` uzyskamy pływanie kolejnych kolumn z lewej strony

# 4 Sposoby wyświetlania elementów

Wszystkie definiowane elementy są wyświetlane na stronie w sposób domyślny. Do zmiany sposobu pokazywania elementu służy atrybut `display`

- `selektor {display: sposób;`
- Parametr `sposób`
  - `block` – element wyświetlany jako element blokowy z odstępami od góra/dół
  - `inline` – element wyświetlany w linii z innymi elementami
  - `list-item` – element wyświetlany jako element wykazu listy `<li>`
  - `none` – element nie będzie wyświetlany
  - `inline-block` – element wyświetlany podobnie jak element zastępowany, mimo że element ma charakter blokowy
  - `run-in` – jeżeli po elemencie występuje element będący blokiem, to ten pierwszy element zostanie wyświetlony w jednej linii z z blokiem

# 4 Sposoby wyświetlania elementów

Elementy blokowe zawsze są wyświetlane od nowej linii i zajmują całą dostępną na stronie szerokość

- Elementami blokowymi są m.in.:
  - `<div>`
  - `<h1>` ... `<h6>`
  - `<p>`
  - `<form>`
  - `<header>`
  - `<footer>`
  - `<section>`

## 4 Sposoby wyświetlania elementów

Elementy wbudowany (InLine) jest wyświetlany w miejscu wstawienia (nie rozpoczyna się od nowej linii) i zajmuje na szerokość tylko tyle miejsca, ile potrzeba na wyświetlenie jego zawartości

- Elementami wbudowanymi są m.in.:
  - `<span>`
  - `<a>`
  - `<img>`

Atrybut `display` może zostać wykorzystany do zamiany sposobu wyświetlania elementów na stronie, czyli elementy InLine można zamieniać na blokowe a blokowe na InLine



## 4 Sposoby wyświetlania elementów

Ukrywanie elementów elementów na stronie można uzyskać po ustawieniu atrybutu `display` na `none`. Innym atrybutem służącym do sterowania wyświetlaniem elementów na stronie jest atrybut `visibility`

- selektor `{visibility: typ;}`
- Wartość parametru `typ`
  - `visible` – element będzie widoczny na ekranie
  - `hidden` – element będzie ukryty (pojaw się pusta przestrzeń)
  - `collapse` – zastosowana dla tabeli ukrywa całą zawartość wiersza lub kolumny, dla innych elementów działa jak `hidden`