



# HTML & CSS & UX

---

Podstawy cz. 3

# Czym jest CSS

Kaskadowe arkusze stylów.

Język opisu formy prezentacji elementów strony internetowej.

- Historia CSS
- Wersje CSS: 1 - 4
- Parsowanie, walidatory, ACID
- Zalety i ograniczenia SASS | LESS

# Rozpoczęcie pracy

Gotowe rozwiązania do usprawnienia procesu tworzenia stron.

- Domyślny arkusz przeglądarki – reset CSS
- CSS Frameworks:  
Bootstrap | Foundation | ...

# Dodawanie deklaracji CSS do dokumentu

- Nagłówek strony HTML PLIK ZEWNĘTRZNY - DOBRA PRAKTYKA

```
<link rel="stylesheet" type="text/css" href="style.css" />
```

- Nagłówek strony HTML DEKLARACJA OSADZONA

```
<style type="text/css"> element {atrybut: wartość;} </style>
```

```
<style type="text/css"><![CDATA[element {atrybut: wartość;}]]></style> <!-- dla dokumentu XHTML -->
```

- Deklaracja na poziomie elementu strony

```
<div style="artybut: wartość; atrybut: wartosc; atrybut: wartość;"></div>
```

# Co można opisać poprzez CSS PODSTAWY

- **jednostki miary elementu:** px, %, em, pt, cm, in ...
- **kolor:** #ff0000, red, rgb(255,0,0), rgba(255,0,0,1) ...
- **rozmiar elementu:** width, height, min-width, max-width, min-height, max-height ...
- **parametry blokowe:** padding (padding-top, padding-left...), margin, border
- **tekst:** font-family, font-size, font-weight, line-height, color, text-align ...
- **parametry tła:** background-color, background-image, background-repeat ...
- **pozycjonowanie i typ\_widoczności:** position, display, overflow

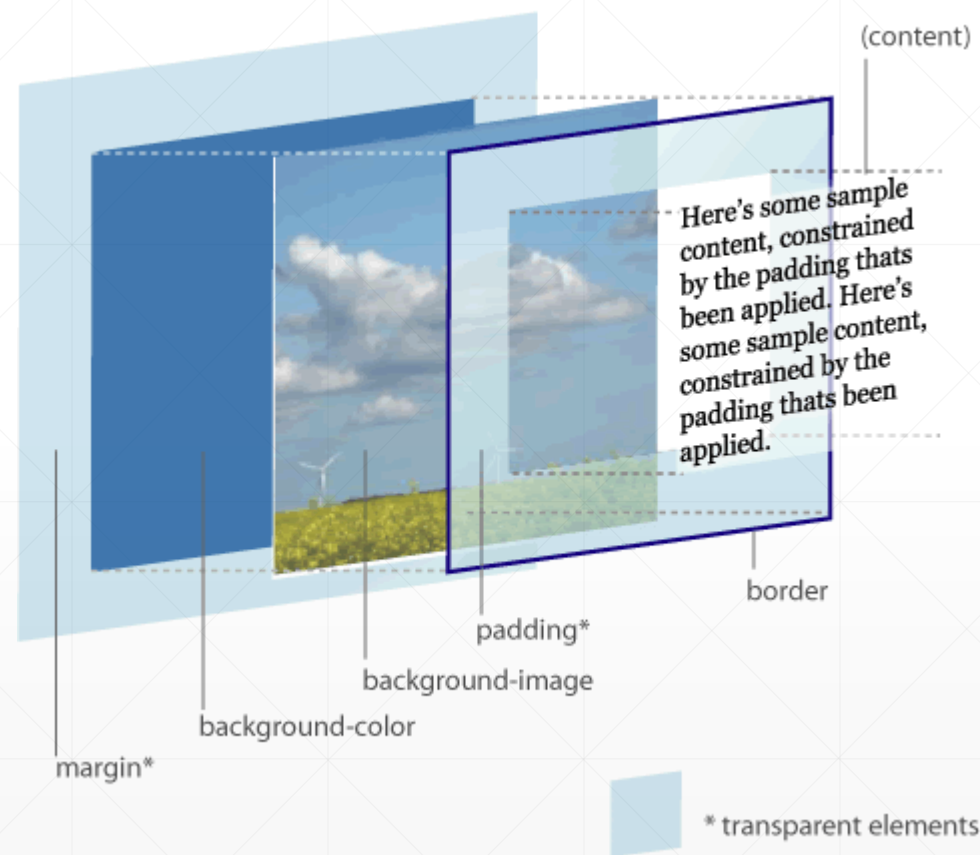
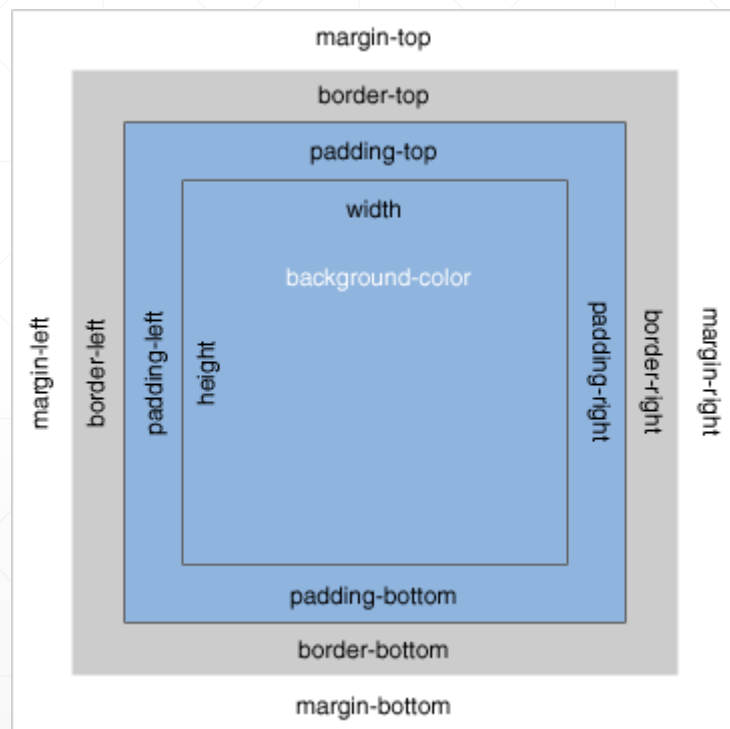
# Skróty w notacji.

- `margin: 5px;`  
`margin-top: 5px; margin-right: 5px; margin-bottom: 5px; margin-left: 5px;`
- `margin: 5px 10px;`  
`margin-top: 5px; margin-right: 10px; margin-bottom: 5px; margin-left: 10px;`
- `margin: 5px 20px 10px;`  
`margin-top: 5px; margin-right: 20px; margin-bottom: 10px; margin-left: 20px;`
- `margin: 10px 20px 30px 40px;`  
`margin-top: 10px, margin-right: 20px, margin-bottom: 30px, margin-left: 40px;`
- `border: solid 3px #f00;`  
`border-width: 3px;`  
`border-color: #f00;`  
`border-style: solid;`

# Elementy tekstowe

- font-family: Arial, sans-serif;
- font-weight: bold; bolder;
- font-size: 10px | 1em | 0.3cm;
- font-style: italic | normal;
- color: red | #ff0000 | #f00 | rgb(255,0,0) | rgba(255,0,0,1);
- text-align: justify, left, right, center;
- text-overflow: clip, ellipsis;

# Model blokowy





# Tło elementu

Kolor:

`background-color: #f00 || red || rgba(255,0,0,1), ....`

Obraz:

`background-image: url('adres_do_obrazka');`

`background-repeat: no-repeat || repeat || repeat-x || repeat-y;`

`background-position: 100px center;`

`background-size: 100px 30% || cover || contain;`

`background-origin: padding-box || border-box || content-box;`

# Zaokrąglanie elementu

`border-radius: 10px, 20px, 30px, 20px;`

# Cień elementu

**box-shadow: 10px 10px 5px 7px #f00 inset;**

box-shadow: przesunięcie x, przesunięcie y, rozmycie, rozmiar, kolor, położenie;

**text-shadow: 10px 10px 5px #f00;**

text-shadow: przesunięcie x, przesunięcie y, rozmycie, kolor;

# Transformacja elementu

`transform:rotate(25deg);`

`transform:translate(50px, 10px);`

`transform:translateX(50px);`

# Tranzycja (efekt prześcia)

**transition: background-color 5s linear 2s;**

transition: właściwość czas-trwania rozkład-w-czasie opóźnienie;

# Filtery

`filter: contrast(200%);`

`filter: grayscale(100%);`

`filter: blur(3px);`

# Kursory

cursor: default || pointer || wait || .....

cursor: url('adres-do-obrazka');



# HTML & CSS & UX

---

Podstawy cz. 4



# Tryb wyświetlania elementów

- Atrybut display:

```
element {  
    display: tryb;  
}
```

gdzie tryb to: block, inline, inline-block, none;

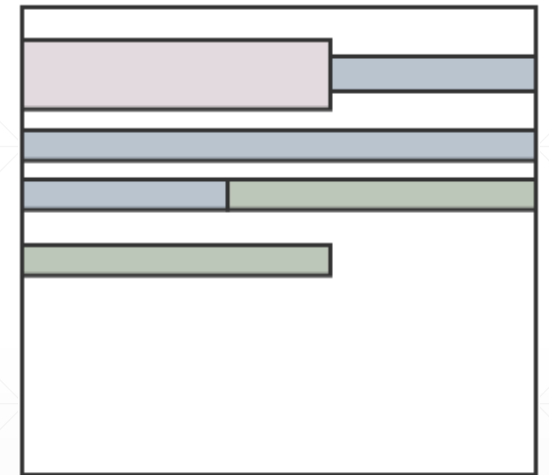
# Typy elementów

- **Elementy liniowe**

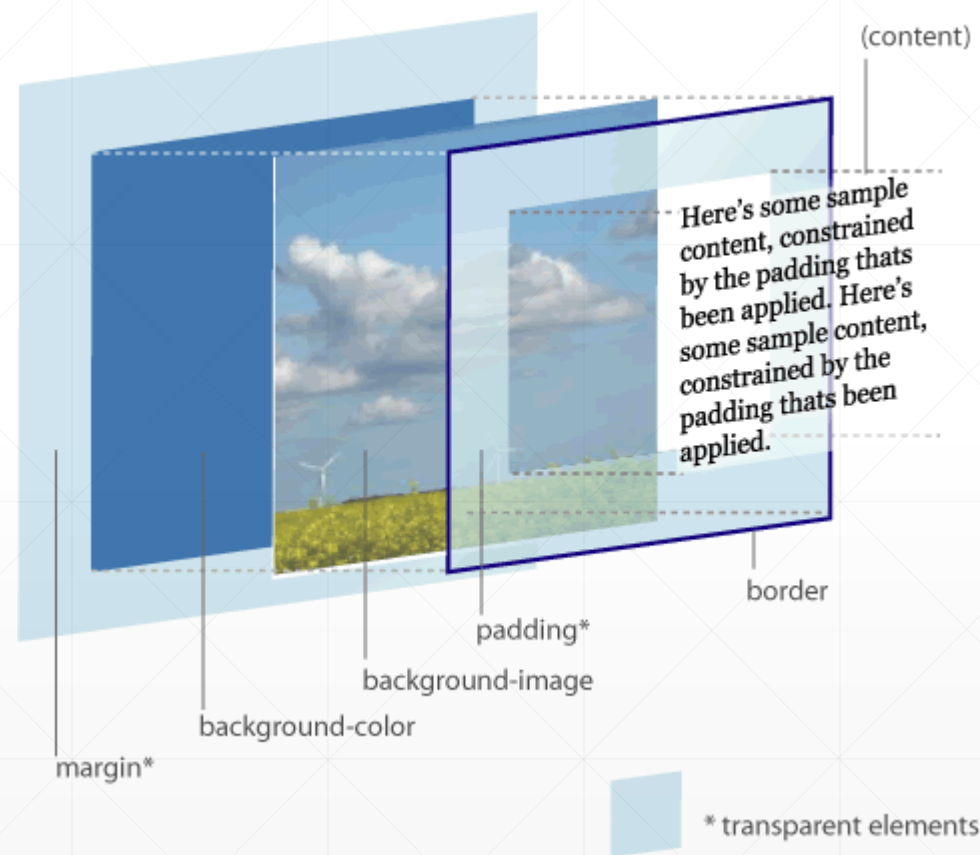
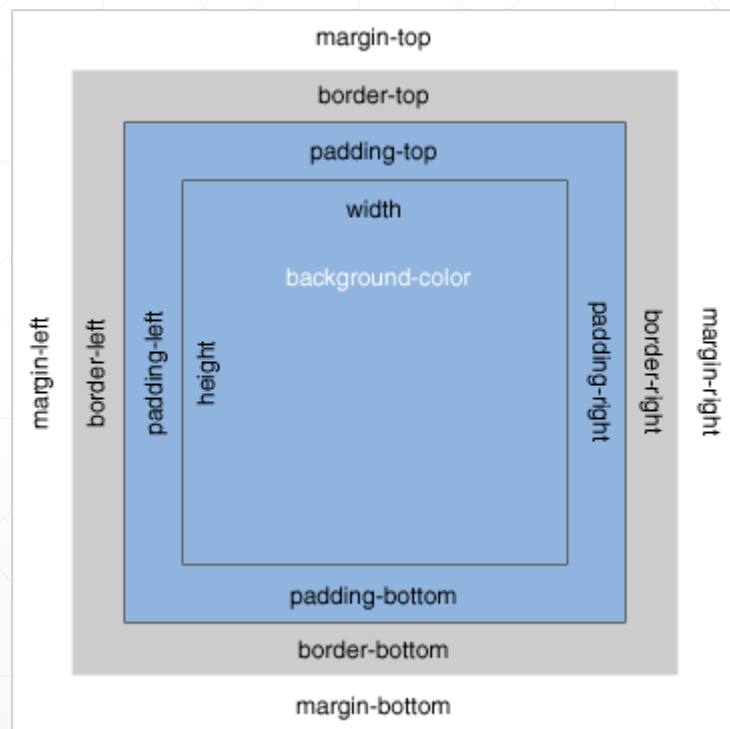
Zajmują wyłącznie przestrzeń wynikającą z zawartej w nich treści.

Mogą zawierać w sobie elementy liniowe lub dane (treść)

`<span>` `<a>` `<input />` `<label>` ...



# Model blokowy



**display: inline**

padding

margin

# Typy elementów

- **Elementy blokowe**

Zajmują całą szerokość elementu nadrzędnego (rodzica).

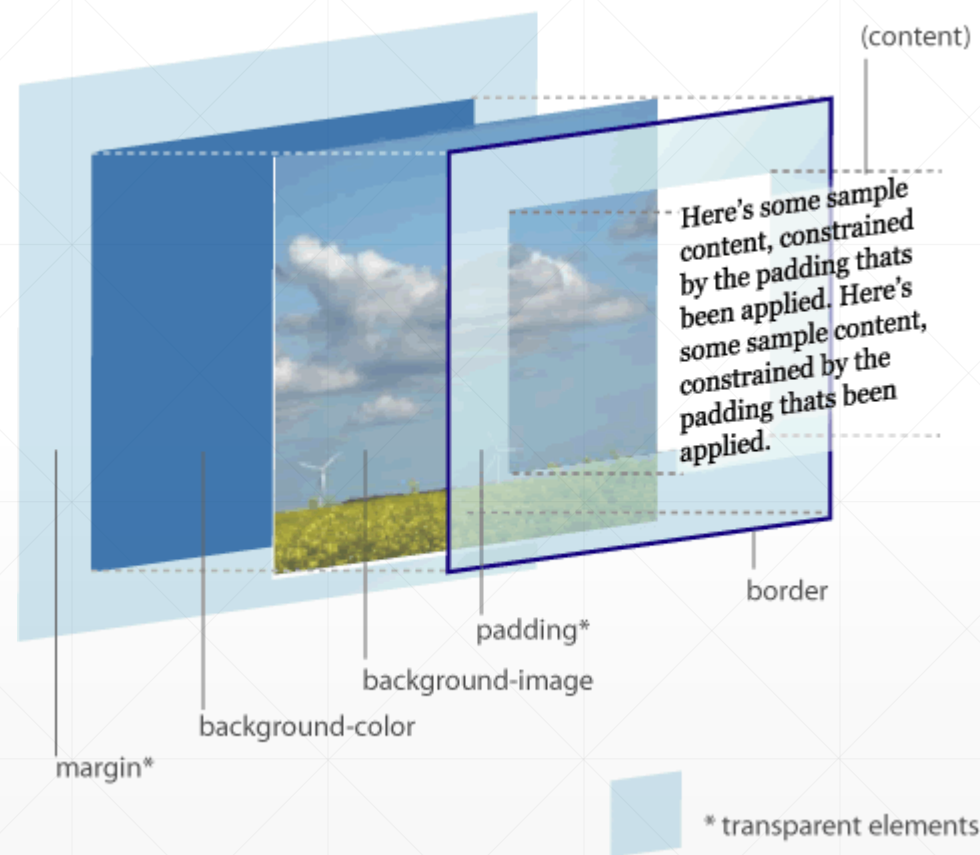
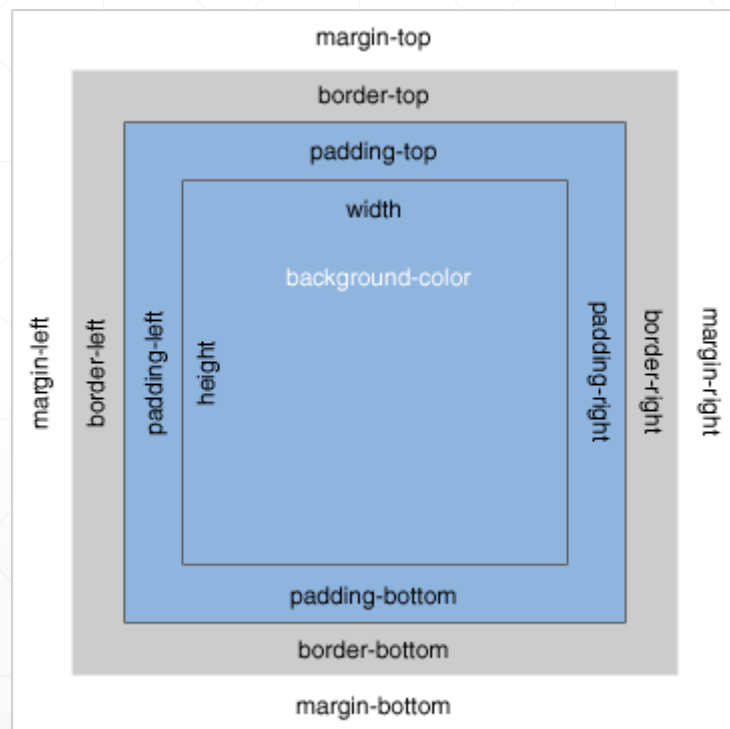
Domyślnie rozpoczynają nową "linię" w treści dokumentu.

Mogą zawierać w sobie elementy blokowe, liniowe lub dane (treść).

`<div>` `<p>` `<form>` `<ul>` `<ol>` `<canvas>` `<table>` `<header>`  
`<article>` `<h1>` `<h2>` ...



# Model blokowy



**display: block**

padding

margin

margin

padding

**display: block**

margin

padding

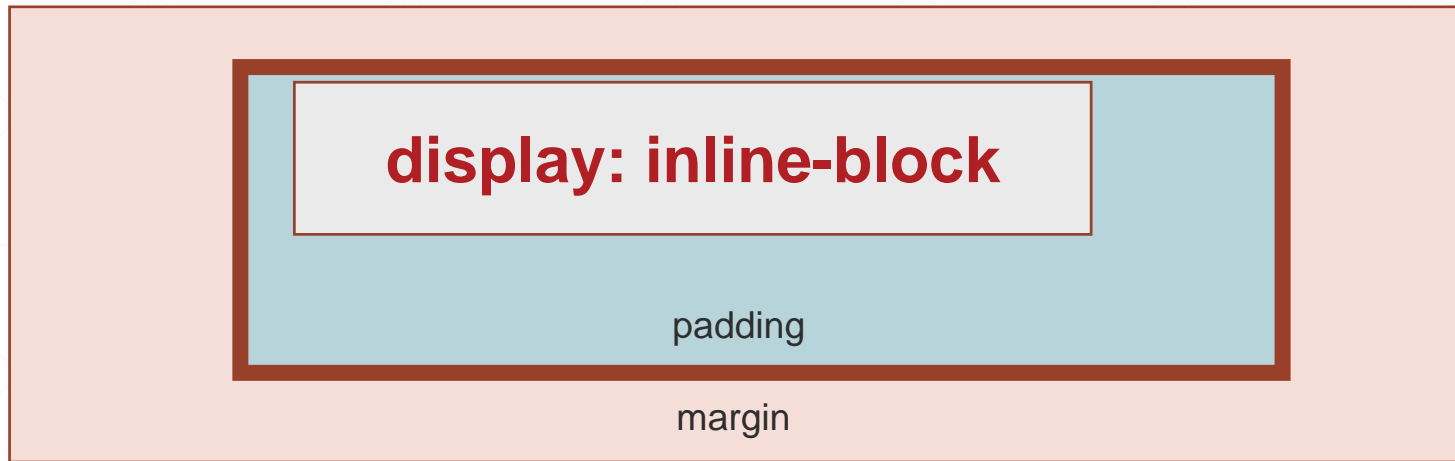
**display: block**

margin

TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT

TEXT TEXT

TEXT TEXT TEXT



TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT

TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT

TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT TEXT





# HTML & CSS & UX

---

Podstawy cz. 5

# Pozycjonowanie – elementy pływające

Atrybut **float: left**, lub **float: right**, wyjmuje element z podstawowego przepływu strony i pozycjonuje go do lewej bądź do prawej krawędzi elementu nadrzednego.

Aby tak wyjęty element umieścić ponownie w przepływie strony należy użyć atrybutu **clear: left** lub **clear: right** dla wybranego typu pozycjonowania, lub atrybutu **clear: both** łącznie dla obu typów pozycjonowania.

Atrybutu clear należy użyć w pierwszym elemencie następującym po elementach wyjętych z przepływu dokumentu.

Dobłą praktyką jest umieszczenie po nich pustego elementu (np `<div>`) z przypisanym atrybutem clear;

# Pozycjonowanie – elementy pływające

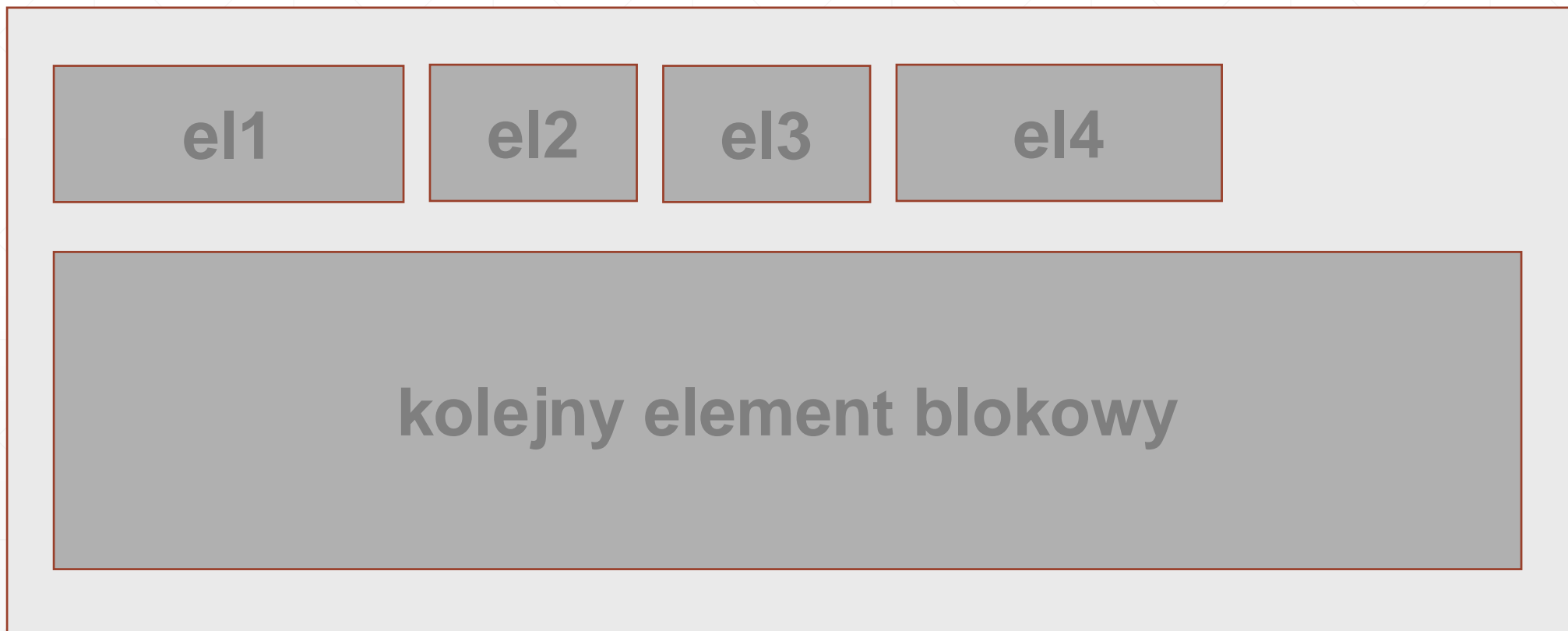
Często spotykaną praktyką jest stworzenie globalnych klas css odpowiedzialnych wyłącznie za zarządzanie elementami pływającymi np:

```
.left {float: left;}
```

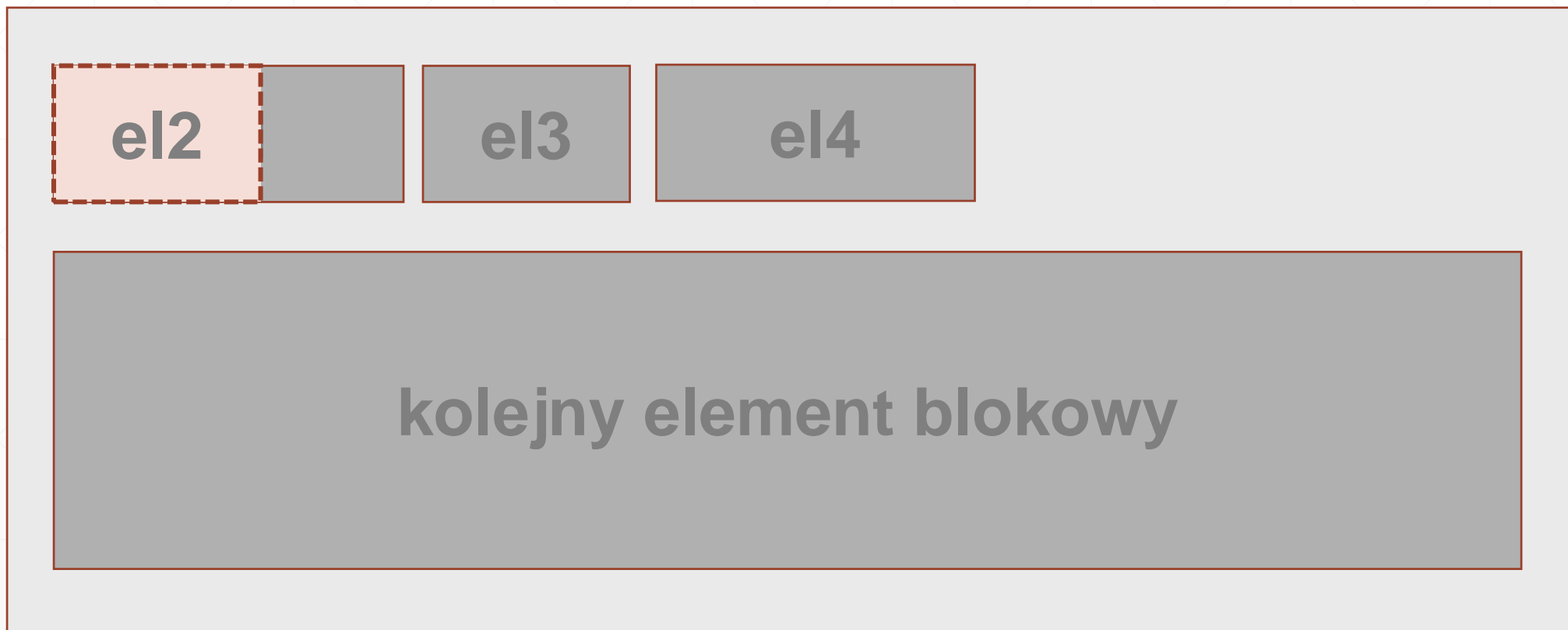
```
.right {float: right;}
```

```
.clear {clear: both;}
```

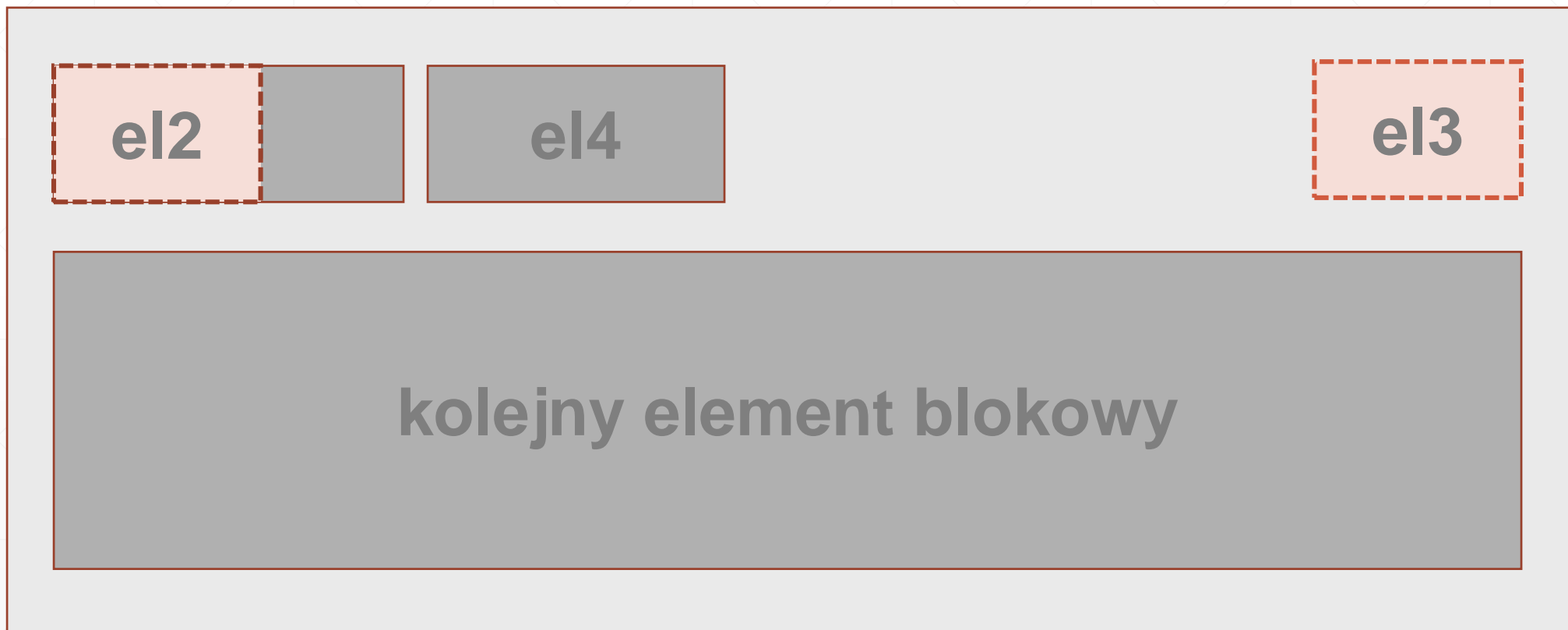
# Pozycjonowanie – elementy pływające



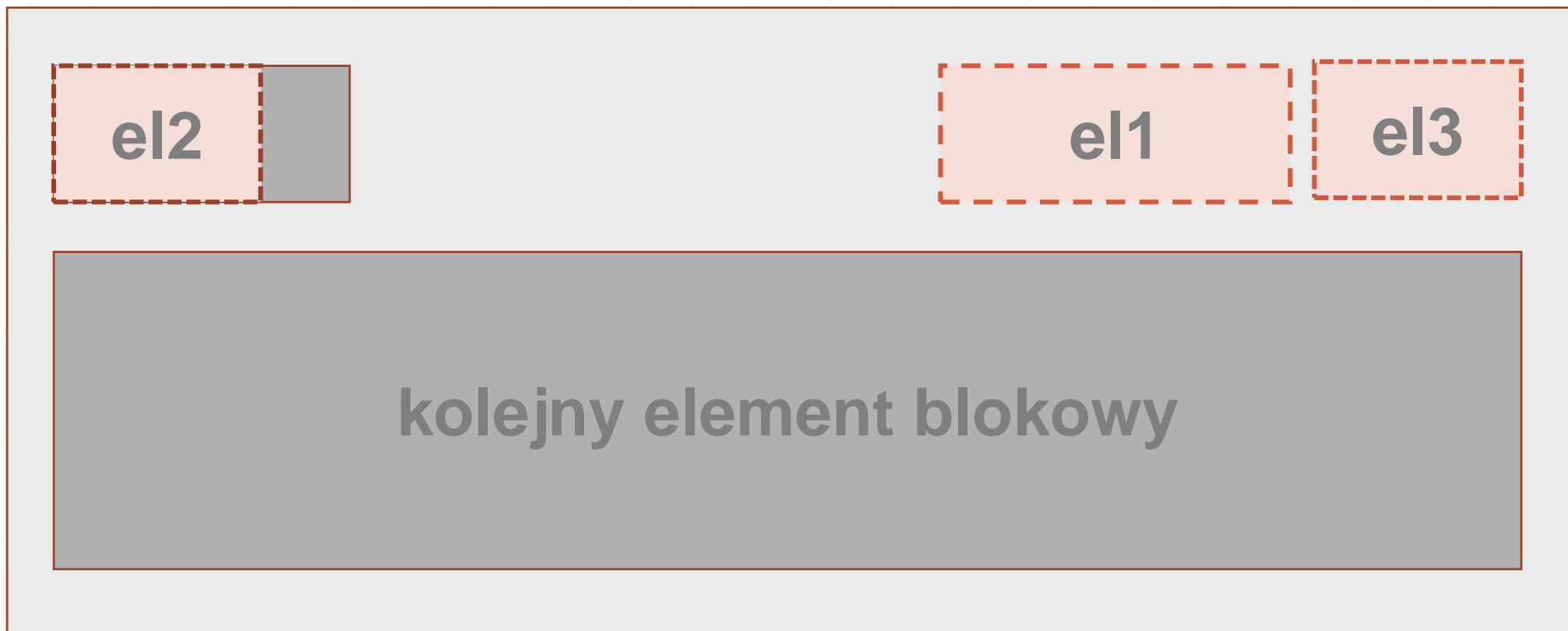
## Pozycjonowanie – elementy pływające



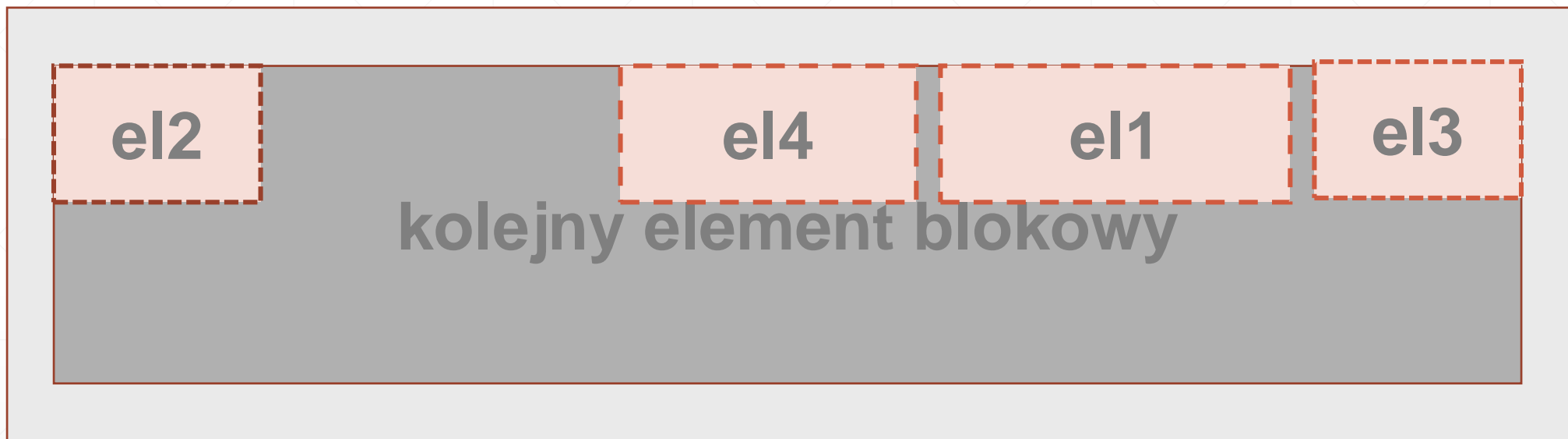
## Pozycjonowanie – elementy pływające



# Pozycjonowanie – elementy pływające

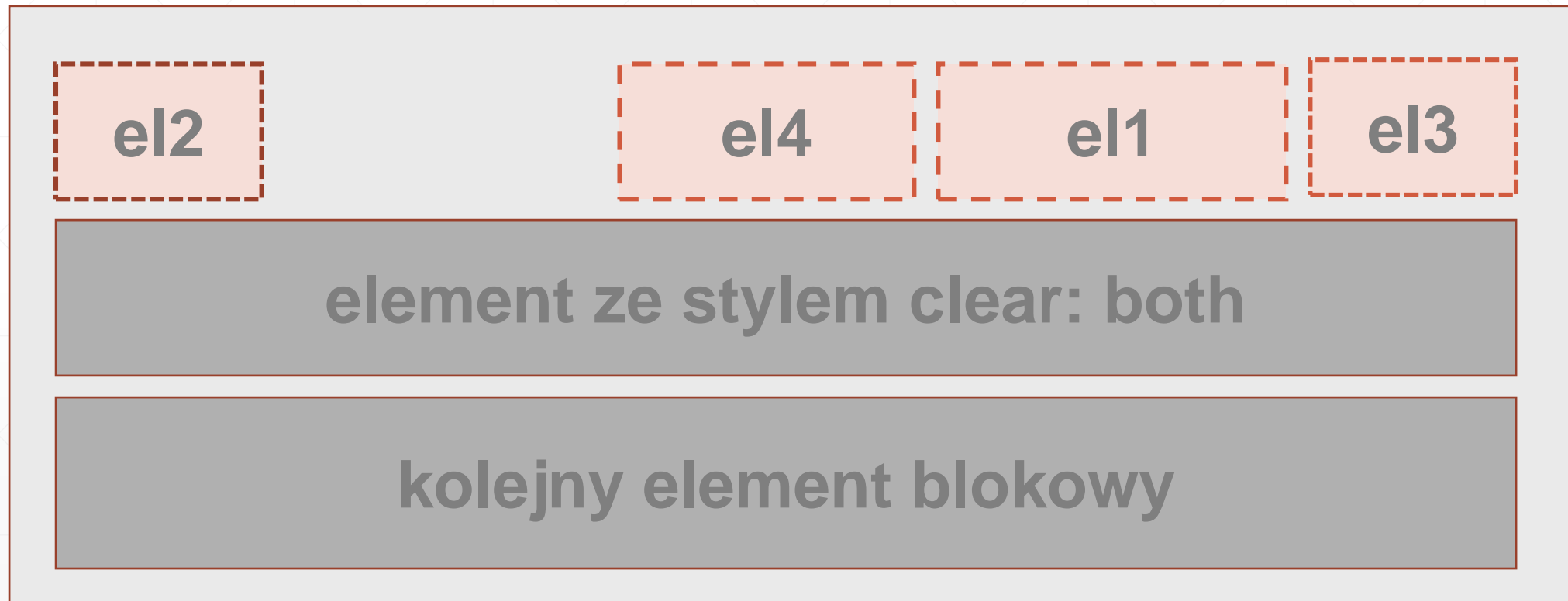


## Pozycjonowanie – elementy pływające





# Pozycjonowanie – elementy pływające



# Pozycjonowanie – relatywne | absolutne | ustalone

Domyślnei elementy HTML wyświetlane są w pozycji statycznej

**position: static;** Możemy zmienić ich pozycję definiując przesunięcie względem:

- .left** – odsunięcie względem lewej krawędzi elementu;
- .right** – odsunięcie względem prawej krawędzi elementu;
- .top** – odsunięcie względem górnej krawędzi elementu;
- .bottom** – odsunięcie względem dolnej krawędzi elementu;

## Pozycjonowanie – relatywne

**position: relative; top: -10px; right: -20px;**

pozycja początkowa

kolejny element

ĆWICZENIE

# Pozycjonowanie – absolutne

**position: absolute; top: -10px; right: 0px;**

pozycja początkowa

kolejny element

**position: relative;**

ĆWICZENIE

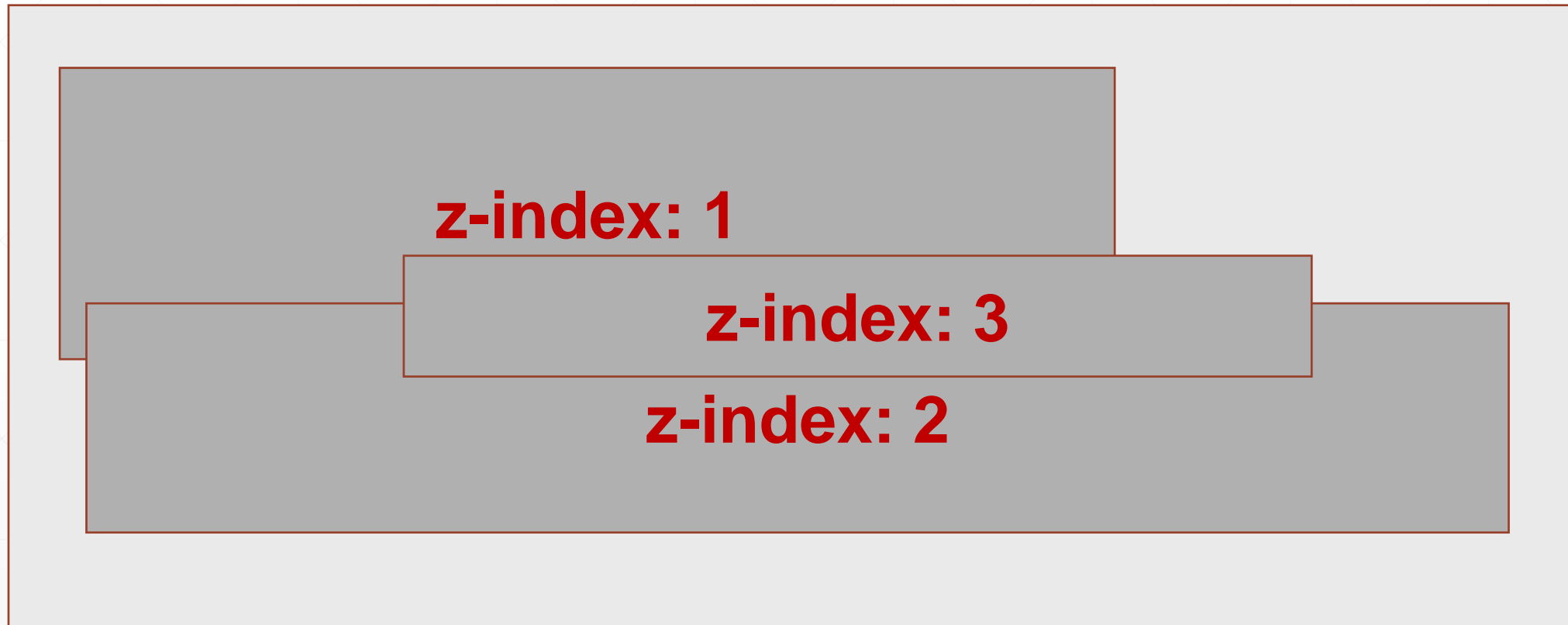
# Pozycjonowanie – absolutne

```
position: absolute;  
top: 10px; right: 10px;  
bottom: 10px; left: 10px;
```

# Pozycjonowanie – fixed



# Pozycjonowanie – stos

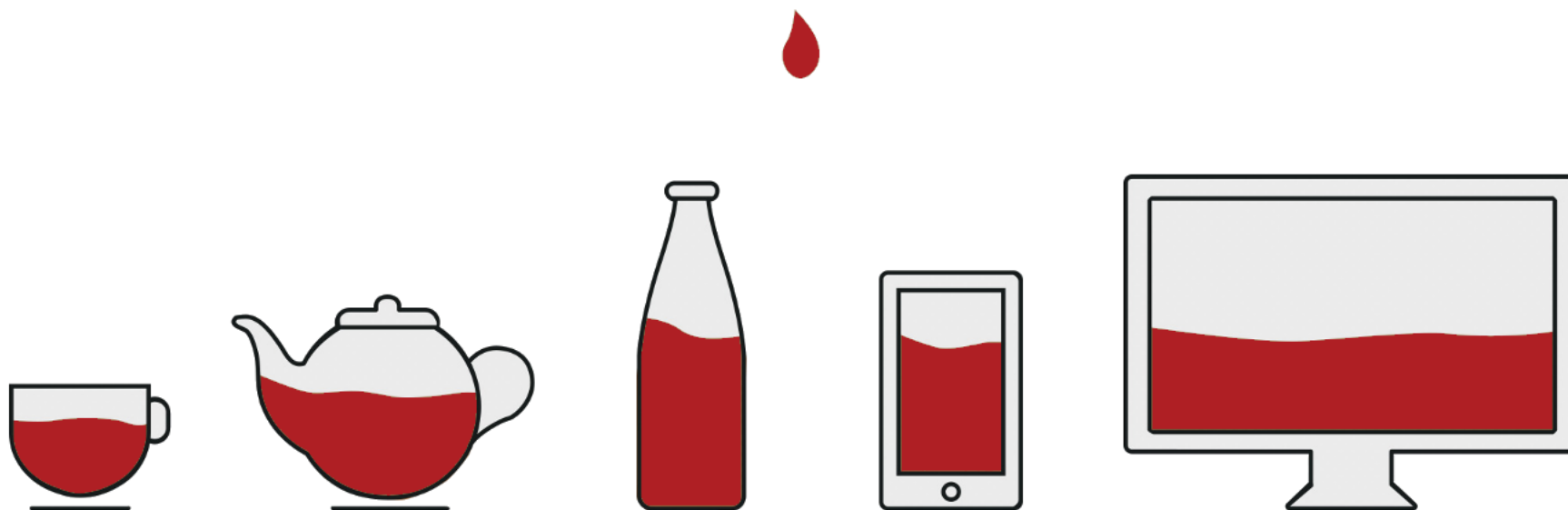


# Układ strony WWW





# CONTENT IS LIKE WATER



“ You put water into a cup it becomes the cup.  
You put water into a bottle it becomes the bottle.  
You put it in a teapot, it becomes the teapot. ”

# CSS Media Queries

```
@media screen {  
  body {  
    background-color: red;  
  }  
}
```

# CSS Media Queries

```
@media screen {  
  body {background-color: red;}  
}
```

```
@media print {  
  body {  
    background-color: none;  
  }  
}
```

# CSS Media Queries

```
@media screen and (max-width: 479px) {  
    aside {display: none;}  
}
```

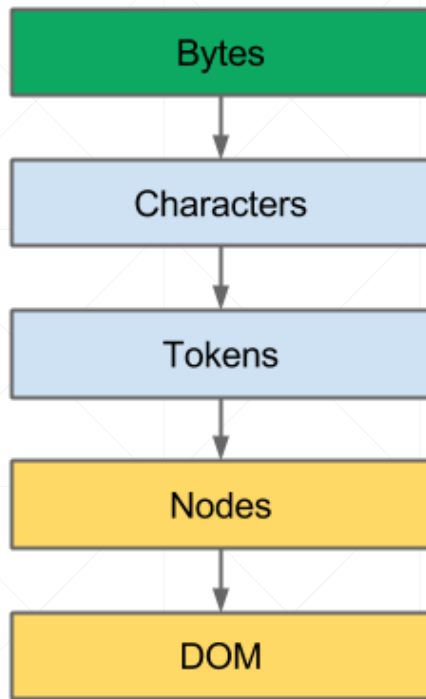
```
@media screen and (min-width: 480px) {  
    aside {display: block;}  
}
```



# HTML & CSS & UX

---

Podstawy cz. 6

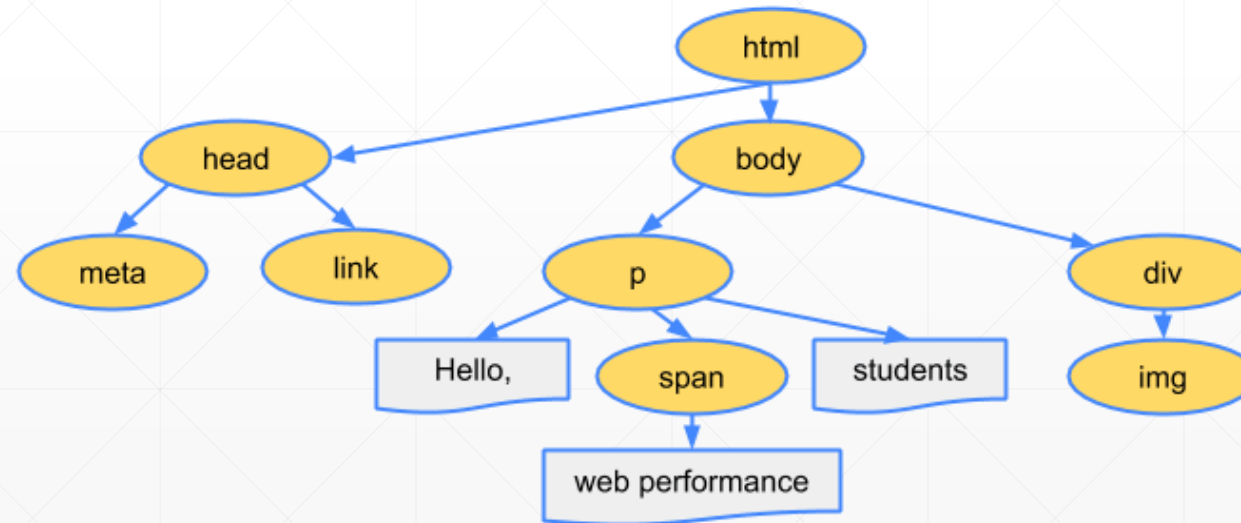


3C 62 6F 64 79 3E 48 65 6C 6C 6F 2C 20 3C 73 70 61 6E 3E 77 6F 72 6C 64 21 3C 2F 73 70 61 6E 3E 3C 2F 62 6F 64 79 3E

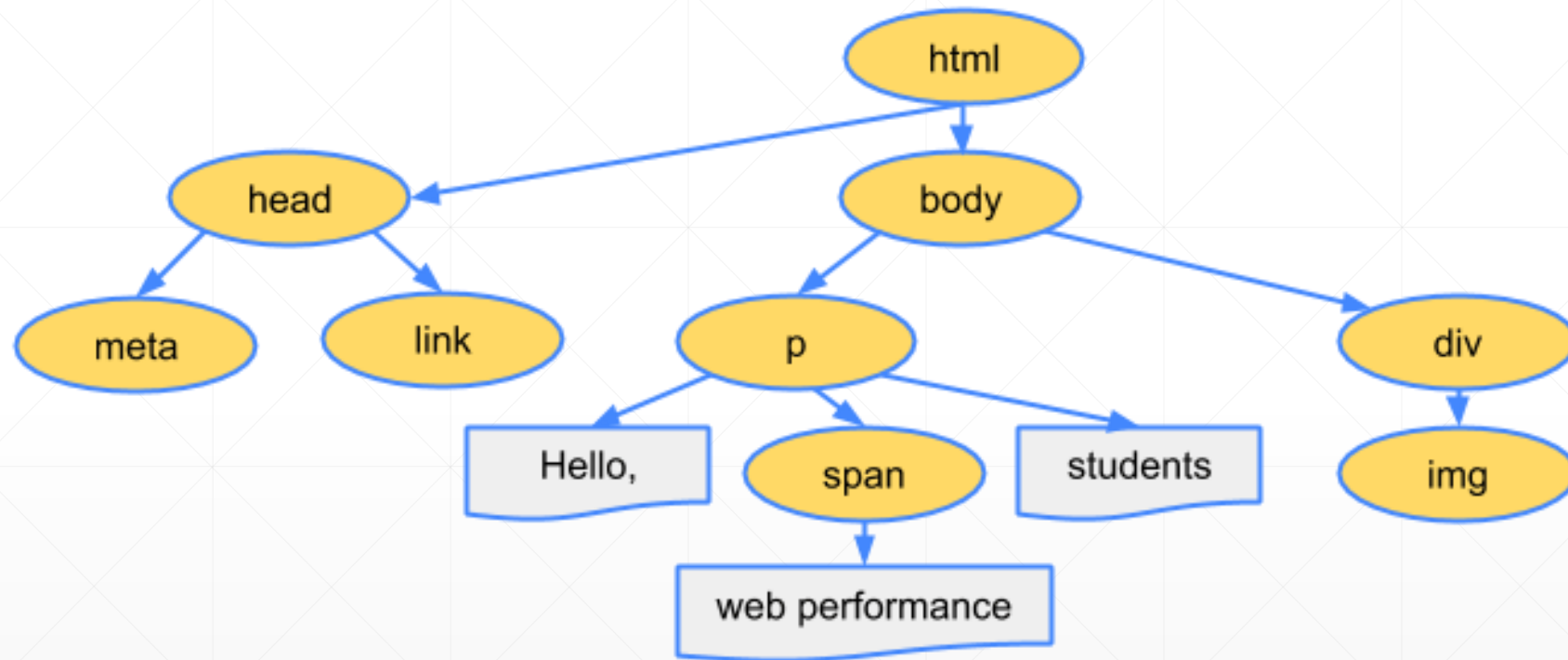
`<html><head>...</head><body><p>Hello <span>web performance</span>...`

StartTag: html   StartTag: head   ...   EndTag: head   StartTag: body   StartTag: p   Hello   ...

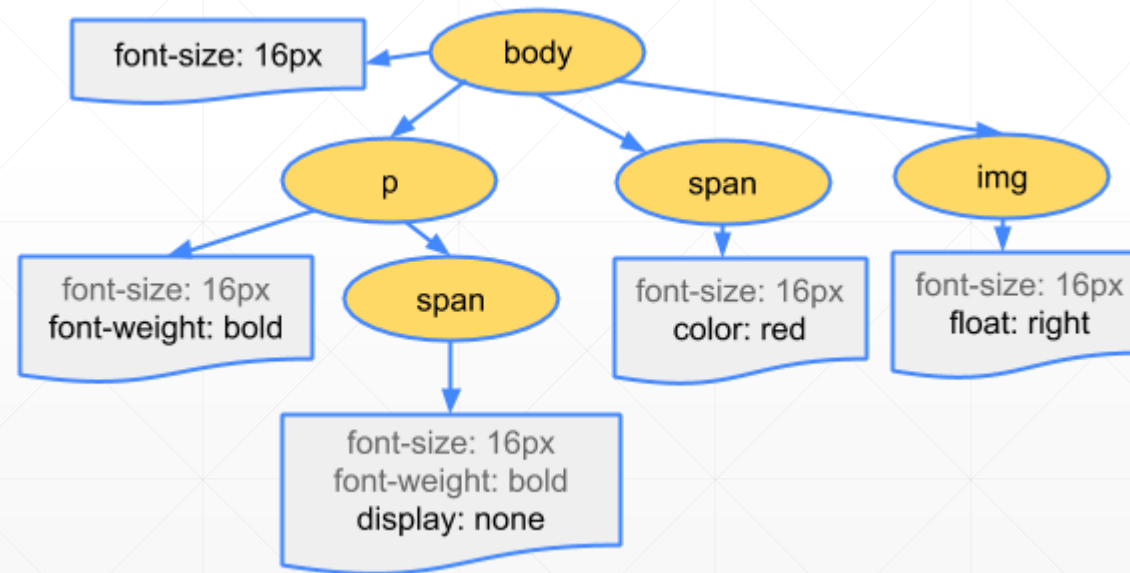
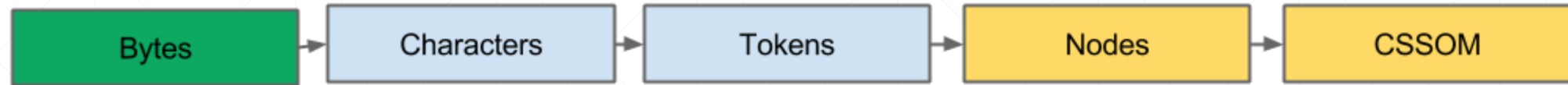
html   head   meta   body   p   Hello



# DOM



# CSSOM







# Model kaskadowy i dziedziczenie

Wszystkie deklaracje uzupełniają się w jeden "wirtualny" styl, który obowiązuje dla strony. Kolejność aplikowania:

- Domyślny arkusz przeglądarki
- Domyślny arkusz użytkownika przeglądarki
- Zewnętrzne arkusze dołączone w sekcji <head> strony
- Deklaracje w atrybucie style="" elementu html

# Model kaskadowy i dziedziczenie

Dane cechy dziedziczą się z elementu rodzica na elementy dzieci.

Dana cecha będzie dziedziczona dopóki kolejny z elementów nie zmodyfikuje jej.

Kaskada i dziedziczenie określają porządek przyjmowania stylów przez elementy. Porządek ten może zostać zmieniony przez dodanie dyrektywy **!important** do wartości danego atrybutu:

```
p {color: red !important;}
```

# Model kaskadowy i dziedziczenie

Wszystkie deklaracje uzupełniają się w jeden "wirtualny" styl, który obowiązuje dla strony. Kolejność aplikowania:

- Domyślny arkusz przeglądarki
- Domyślny arkusz użytkownika przeglądarki
- Zewnętrzne arkusze dołączone w sekcji <head> strony
- Deklaracje w atrybucie style="" elementu html

# Model kaskadowy i dziedziczenie

Dane cechy dziedziczą się z elementu rodzica na elementy dzieci.

Dana cecha będzie dziedziczona dopóki, któryś z elementów nie zmodyfikuje jej.

Kaskada i dziedziczenie określają porządek przyjmowania stylów przez elementy. Porządek ten może zostać zmieniony przez dodanie dyrektywy **!important** do wartości danego atrybutu:

```
p {color: red !important;}
```

## Selektory – uzupełnianie się deklaracji

```
div {color: red; font-weight: bold !important;}
```

```
div {color: blue; font-size: 10px; font-weight: bolder;}
```

```
div {  
    color: blue;  
    font-weight: bold !important;  
    font-size: 10px;  
}
```

# Selektory – uzupełnianie się deklaracji

StylesComputedEvent ListenersDOM BreakpointsProperties

Filter

element.style {  
}

body, .g, html, .std, .c h2, h1, #cdr\_min, #cdr\_max, .cpbb, .kpbb, .kprb, .kpgb, .kpgrb, .ksb {  
font-family: arial,sans-serif;  
}

body {  
color: ■ #222;  
}

.g, body, html, input, .std, h1 {  
font-size: small;  
font-family: arial,sans-serif;  
}

body {  
background: ▢ #fff;  
}

body {  
color: ■ #000;  
margin: 0;  
overflow-y: scroll;  
}

body {  
display: block;  
margin: 8px;  
}

Inherited from html

webhp?sourceid=...=2&ie=UTF-8:14

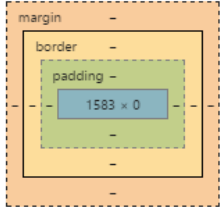
webhp?sourceid=...=2&ie=UTF-8:14

webhp?sourceid=...=2&ie=UTF-8:14

webhp?sourceid=...=2&ie=UTF-8:14

user agent stylesheet

StylesComputedEvent ListenersDOM BreakpointsProperties



Filter

background-attachmentscroll

background-clipborder-box

background-color▢ rgb(255, 255, 255)

background-imagenone

background-originpadding-box

background-sizeauto

color■ rgb(34, 34, 34)

displayblock

font-familyarial, sans-serif

font-size13px

small.g, body, html, input, .std, h1webhp?sourceid=...=2&ie=UTF-8:14

small.g, body, html, input, .std, h1webhp?sourceid=...=2&ie=UTF-8:14

height0px

# Rozpoczęcie pracy

Gotowe rozwiązania do usprawnienia procesu tworzenia stron.

- Domyślny arkusz przeglądarki – reset CSS
- CSS Frameworks:  
Bootstrap | Foundation | ...
- **kolejność ładowania plików CSS**
- **cache przeglądarki**



# Selektory CSS

Let's eat grandpa.  
Let's eat, grandpa.

**correct punctuation can  
save a person`s life.**

# Selektory – typ elementu

```
h1 {  
    atrybut: wartość;  
}
```

## Selektory – klasa elementu

```
.nazwa-klasy {  
    atrybut: wartość;  
}
```

## Selektory – identyfikator elementu

```
#identyfikatorElementu {  
    atrybut: wartość;  
}
```

## Selektory – atrybut

```
[atributElementuHTML="wartość"] {  
    atybut:wartość;  
}
```

```
img[title="opis"] {...}  
img[title^="opis"] {...}  
img[title*="opis"] {...}
```

## Selektory – pseudoklasy

```
SELEKTOR:pseudoklasa {  
    atrybut: wartość;  
}
```

## Selektory – pseudoklasy

```
div:hover {  
    background-color: #f00;  
}
```

// zmień kolor tła elementu div po najechaniu kursorem

## Selektory – łączenie selektorów

**div.nazwa-klasy** {atrybut: wartość;}

**#identyfikatorElementy.nazwa-klasy** {...}

**.nazwa-klasy.inna-nazwa-klasy** {...}

**p#idEl.nazwa-klasy.c2.c3:hover** {...}



## Selektory – wspólna definicja

**.nazwa-klasy,**  
**.inna-nazwa-klasy,**  
**#identyfikatorElementu: hover,**  
**div {atrybut: wartość;}**

# Selektory – pseudoklasy i pseudoelementy

```
div:hover {  
    background-color: #f00;  
}
```

// zmień kolor tła elementu div po najechaniu kursorem

## Selektory – wybrane pseudoklasy

**Pseudoklasy linków:** :link, :visited

**Pseudoklasy akcji:** :active, :hover, :focus

**Pseudoklasy interfejsu:** :enabled, :disabled, :checked

**Pseudoklasy strukturalne:** :nth-child(), :nth-last-child(), :nth-of-type(), :nth-last-of-type(), :first-child, :last-child, :only-child, :first-of-type, :last-of-type, :only-of-type, :empty

**Pseudoklasa negacji:** :not()

# Selektory – wybrane pseudoelementy

**:first-line**

**:first-letter**

**:before**

**:after**

```
#nawigacja a[href^='http://']::after {  
  content: 'link zewnętrzny';  
  color: red;  
}
```

# Selektory – precyzyjne zapytania

**h1 {...}**

**header h1 {...}**

# Selektory – precyzyjne zapytania

```
<header id="naglowek-artykulu">  
  <h1 class="tytul-artykulu">...</h1>  
</header>
```

**header h1 {...}**

**#naglowek-artykulu h1 {...}**

**#naglowek-artykulu h1.tytul-artykulu {...}**

**#naglowek-artykulu .tytul-artykulu {...}**

# Selektory – precyzyjne zapytania

**#naglowek nav.navigacja ul > li:nth-child(2n):hover span { }**

# Waga selektorów



**a**

1 x element selector

Sith: 0, 0, 1



**p a**

2 x element selectors

Sith: 0, 0, 2



**.whatever**

1 x class selector

Sith: 0, 1, 0



**a.whatever**

1 x element selector  
1 x class selector

Sith: 0, 1, 1



**p a.whatever**

2 x element selectors  
1 x class selector

Sith: 0, 1, 2



**.whatever .whatever**

2 x class selectors

Sith: 0, 2, 0



**p.whatever a.whatever**

2 x element selectors  
2 x class selectors

Sith: 0, 2, 2



**#whatever**

1 x id selector

Sith: 1, 0, 0



**a#whatever**

1 x element selector  
1 x id selector

Sith: 1, 0, 1



**.whatever a#whatever**

1 x element selectors  
1 x class selector  
1 x id selector

Sith: 1, 1, 1



**.whatever .whatever #whatever**

2 x class selectors  
1 x id selector

Sith: 1, 2, 0



**#whatever #whatever**

2 x id selectors

Sith: 2, 0, 0





# HTML & CSS & UX

---

Podstawy cz. 7