



# Jednostki w RWD

# Em a font-size

## Em

Zaczniemy od tego, że użycie `em` w `font-size`, odnosi się do najbliższego rodzica a nie do tego samego elementu. Czyli wielkość czcionki jest obliczana na podstawie czcionki rodzica.

Zobacz przykład:

## HTML

```
<div class="parent">
  Jestem rodzicem
  <div class="child">
    Jestem dzieckiem
  </div>
</div>
```

## CSS

```
.parent {
  font-size: 10px;
}
```

```
.child {
  font-size: 2em;
}
```

# Em a font-size

## Em

Zaczniemy od tego, że użycie `em` w `font-size`, odnosi się do najbliższego rodzica a nie do tego samego elementu. Czyli wielkość czcionki jest obliczana na podstawie czcionki rodzica.

Zobacz przykład:

## HTML

```
<div class="parent">
  Jestem rodzicem
  <div class="child">
    Jestem dzieckiem
  </div>
</div>
```

## CSS

```
.parent {
  font-size: 10px;
}
```

Jestem rodzicem - 10px.

```
.child {
  font-size: 2em;
}
```

# Em a font-size

## Em

Zaczniemy od tego, że użycie `em` w `font-size`, odnosi się do najbliższego rodzica a nie do tego samego elementu. Czyli wielkość czcionki jest obliczana na podstawie czcionki rodzica.

Zobacz przykład:

## HTML

```
<div class="parent">
  Jestem rodzicem
  <div class="child">
    Jestem dzieckiem
  </div>
</div>
```

## CSS

```
.parent {
  font-size: 10px;
}
```

Jestem rodzicem - 10px.

```
.child {
  font-size: 2em;
}
```

Jestem dzieckiem - **20px** (bo 10px x 2).

# Em a inne wielkości

## Em

Ale definicja em mówi, że jest to jednostka relatywna odnosząca się do wielkości czcionki tego elementu, na którym jest użyta!

Zobacz przykład:

## HTML

```
<div class="parent">
  Jestem rodzicem
  <div class="child">
    Jestem dzieckiem
  </div>
</div>
```

## CSS

```
.parent {
  font-size: 10px;
}
```

```
.child {
  font-size: 2em;
  width: 2em;
  height: 4em;
  border: 1px solid red;
}
```

# Em a inne wielkości

## Em

Ale definicja em mówi, że jest to jednostka relatywna odnosząca się do wielkości czcionki tego elementu, na którym jest użyta!

Zobacz przykład:

## HTML

```
<div class="parent">
  Jestem rodzicem
  <div class="child">
    Jestem dzieckiem
  </div>
</div>
```

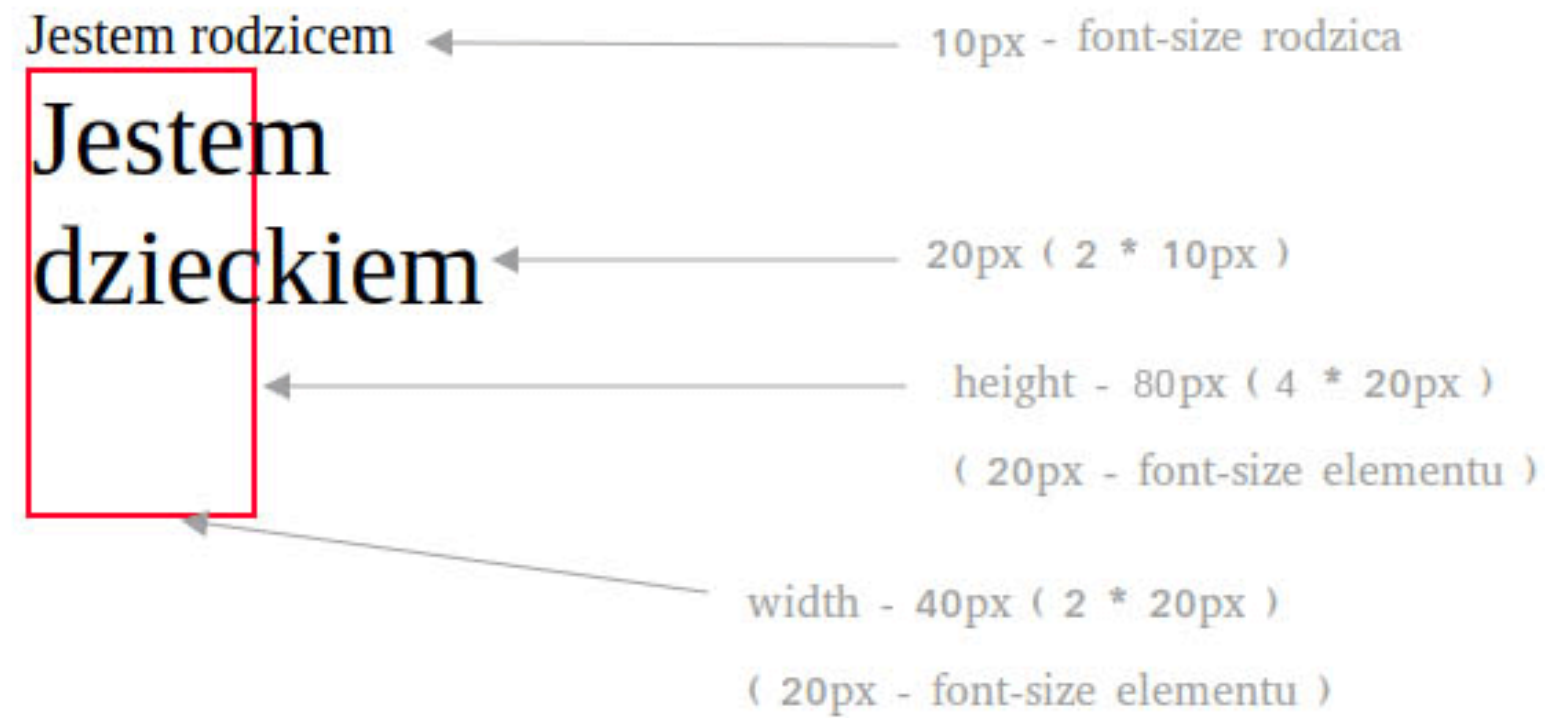
## CSS

```
.parent {
  font-size: 10px;
}
```

Jestem rodzicem - 10px.

```
.child {
  font-size: 2em;
  width: 2em;
  height: 4em;
  border: 1px solid red;
}
```

# Em - podsumowanie



## Podsumowując em

Jednostka ta odnosi się do wielkości czcionki danego elementu czyli wszystkie wielkości ustawione w em są obliczane na podstawie własności `font-size` tego elementu.

**Wyjątkiem** jest samo ustawianie `font-size`. W takim przypadku wielkość jest obliczana na podstawie rodzica.

# Rem a font-size

## Rem

Jednostka `rem` jest również jednostką relatywną. Jest natomiast obliczana na podstawie wielkości czcionki użytej w elemencie `root`, którym zazwyczaj jest element `html`.

## HTML

```
<div class="parent">
  Jestem rodzicem
  <div class="child">
    Jestem dzieckiem
  </div>
</div>
```

## SCSS

```
html {
  font-size: 20px;
}
.parent {
  font-size: 10px;
}
.child {
  font-size: 2rem;
  width: 2rem;
  border: 1px solid red;
}
```



# Rem a font-size

## Rem

Jednostka `rem` jest również jednostką relatywną. Jest natomiast obliczana na podstawie wielkości czcionki użytej w elemencie `root`, którym zazwyczaj jest element `html`.

## HTML

```
<div class="parent">
  Jestem rodzicem
  <div class="child">
    Jestem dzieckiem
  </div>
</div>
```

## SCSS

```
html {
  font-size: 20px;
}
.parent {
  font-size: 10px;
}
.child {
  font-size: 2rem;
  width: 2rem;
  border: 1px solid red;
}
```

Ten element będzie rootem.

# Rem a font-size

## Rem

Jednostka `rem` jest również jednostką relatywną. Jest natomiast obliczana na podstawie wielkości czcionki użytej w elemencie `root`, którym zazwyczaj jest element `html`.

## HTML

```
<div class="parent">
  Jestem rodzicem
  <div class="child">
    Jestem dzieckiem
  </div>
</div>
```

## SCSS

```
html {
  font-size: 20px;
}
.parent {
  font-size: 10px;
}
.child {
  font-size: 2rem;
  width: 2rem;
  border: 1px solid red;
}
```

Ten element będzie rootem.  
Jestem rodzicem - `10px`.

# Rem a font-size

## Rem

Jednostka `rem` jest również jednostką relatywną. Jest natomiast obliczana na podstawie wielkości czcionki użytej w elemencie `root`, którym zazwyczaj jest element `html`.

## HTML

```
<div class="parent">
  Jestem rodzicem
  <div class="child">
    Jestem dzieckiem
  </div>
</div>
```

## SCSS

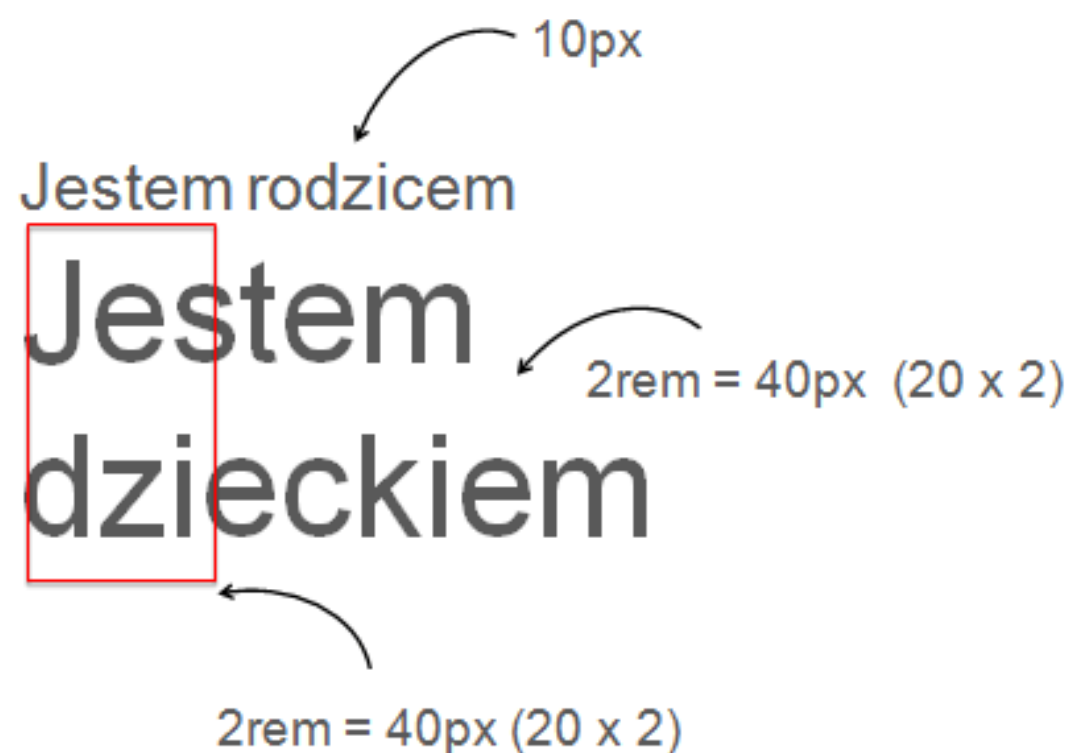
```
html {
  font-size: 20px;
}
.parent {
  font-size: 10px;
}
.child {
  font-size: 2rem;
  width: 2rem;
  border: 1px solid red;
}
```

Ten element będzie rootem.  
Jestem rodzicem - `10px`.  
Ten element będzie dzieckiem - `40px`  
(bo  $20 \times 2$ ).

# Rem a inne wielkości

## Rem

Podobnie jeśli chodzi o ustawianie innych wielkości w rem, również wartości są obliczane na podstawie wielkości czcionki ale z elementu root.



## HTML

```
<div class="parent">
  Jestem rodzicem
  <div class="child">
    Jestem dzieckiem
  </div>
</div>
```

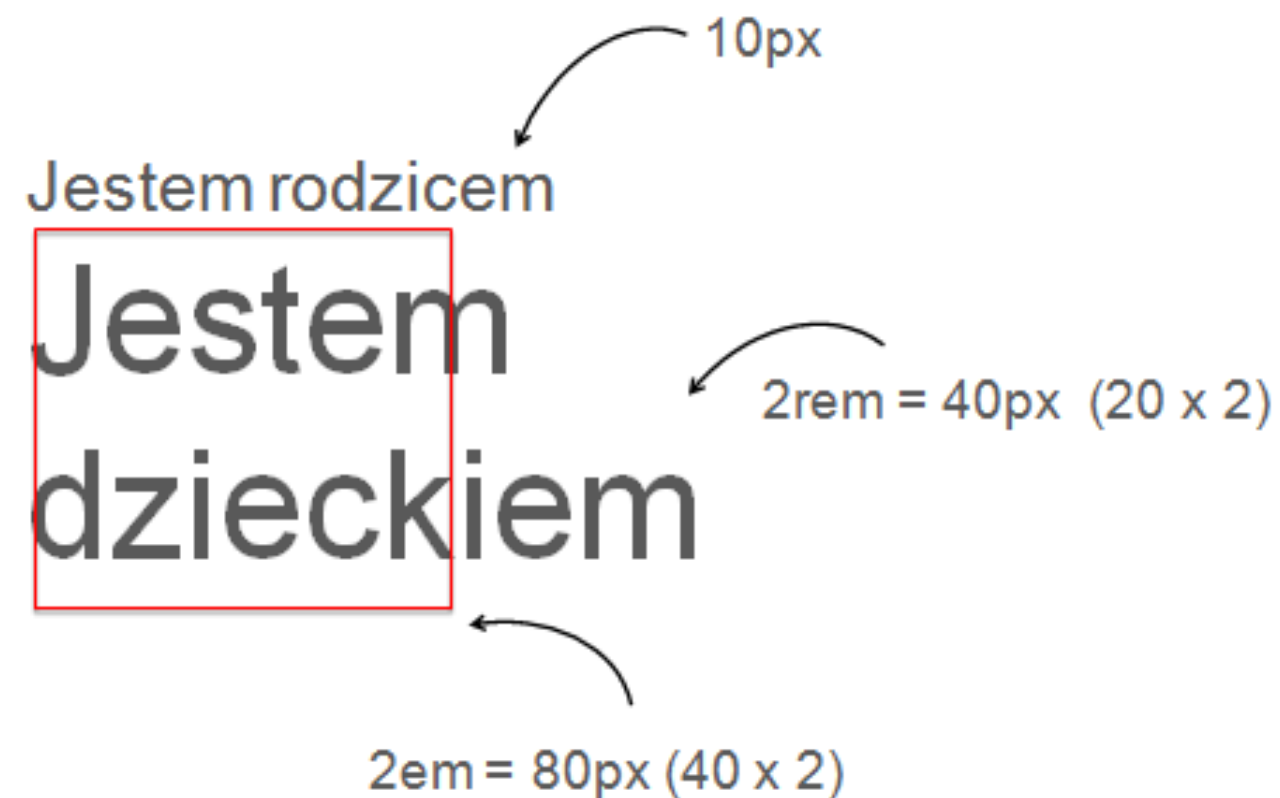
## CSS

```
html {
  font-size: 20px;
}
.parent {
  font-size: 10px;
}
.child {
  font-size: 2rem;
  width: 2rem;
  border: 1px solid red;
}
```

# Em razem z Rem

## Em użyte razem z rem

Tutaj zasada się nie zmienia. Em w przypadku innych właściwości niż font-size będzie obliczany na podstawie wielkości czcionki danego elementu, nie ważne czy jest ona ustawiona za pomocą px czy rem.



## HTML

```
<div class="parent">  
  Jestem rodzicem  
  <div class="child">  
    Jestem dzieckiem  
  </div>  
</div>
```

## CSS

```
html {  
  font-size: 20px;  
}  
.parent {  
  font-size: 10px;  
}  
.child {  
  font-size: 2rem;  
  width: 2em;  
  border: 1px solid red;  
}
```

# Procent – font-size

## %

Jednostka % ustawia wielkość na podstawie najbliższego przodka - jeśli element parent ma ustawioną czcionkę na **10px** to element children z ustawioną czcionką na **200%** będzie ją miał 2 razy większą czyli **20px**.

## HTML

```
<div class="parent">
  Jestem rodzicem
  <div class="child">
    Jestem dzieckiem
  </div>
</div>
```

## CSS

```
html {
  font-size: 20px;
}
.parent {
  font-size: 10px;
}
.child {
  font-size: 200%;
  width: 2rem;
  border: 1px solid red;
}
```

# Procent – font-size

## %

Jednostka % ustawia wielkość na podstawie najbliższego przodka - jeśli element parent ma ustawioną czcionkę na **10px** to element children z ustawioną czcionką na **200%** będzie ją miał 2 razy większą czyli **20px**.

## HTML

```
<div class="parent">
  Jestem rodzicem
  <div class="child">
    Jestem dzieckiem
  </div>
</div>
```

## CSS

```
html {
  font-size: 20px;
}
.parent {
  font-size: 10px;
}
.child {
  font-size: 200%;
  width: 2rem;
  border: 1px solid red;
}
```

Jestem rodzicem - **10px**.

# Procent – font-size

## %

Jednostka % ustawia wielkość na podstawie najbliższego przodka - jeśli element parent ma ustawioną czcionkę na **10px** to element children z ustawioną czcionką na **200%** będzie ją miał 2 razy większą czyli **20px**.

## HTML

```
<div class="parent">
  Jestem rodzicem
  <div class="child">
    Jestem dzieckiem
  </div>
</div>
```

## CSS

```
html {
  font-size: 20px;
}
.parent {
  font-size: 10px;
}
.child {
  font-size: 200%;
  width: 2rem;
  border: 1px solid red;
}
```

Jestem rodzicem - **10px**.

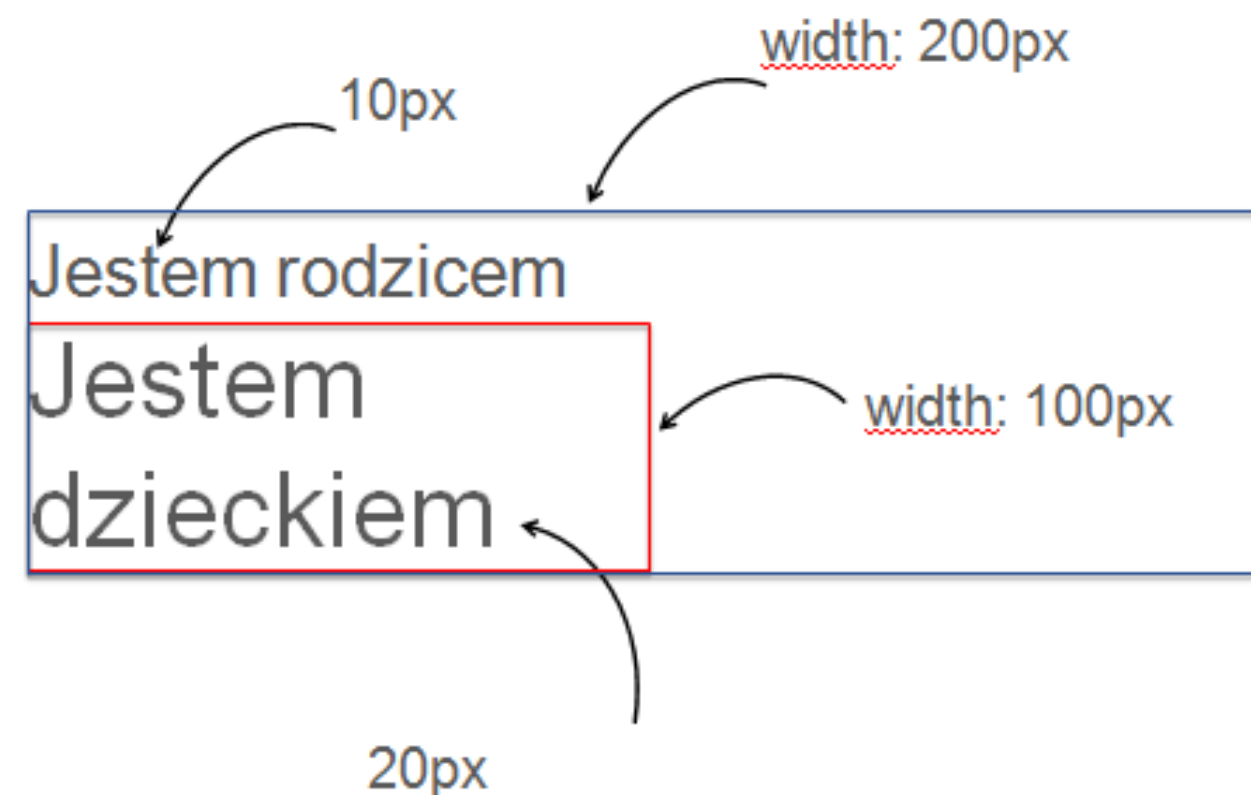
Jestem dzieckiem - **20px** (bo 10 x 2).



# Procent – width

## Określenie width w procentach

W przypadku width również wielkość jest obliczana na podstawie rodzica. Na przykładzie obok widać, że element parent ma ustawioną szerokość na 200px, a dziecko na 50%, co po obliczeniu daje 100px.



## HTML

```
<div class="parent">
  Jestem rodzicem
  <div class="child">
    Jestem dzieckiem
  </div>
</div>
```

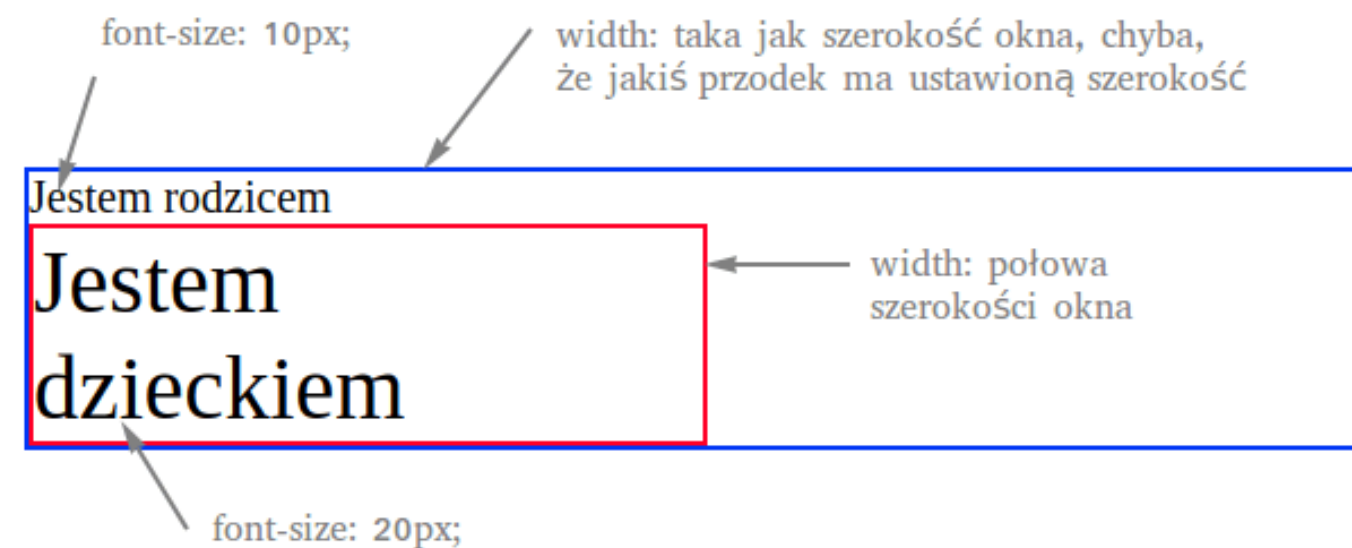
## CSS

```
.parent {
  font-size: 10px;
  width: 200px;
  border: 1px solid blue;
}
.child {
  font-size: 200%;
  width: 50%;
  border: 1px solid red;
}
```

# Procent – width

## Określenie width w procentach

Jeśli nie ustawisz elementowi parent żadnej szerokości to będzie on zajmował całą dostępną szerokość rodzica (w naszym przypadku okno przeglądarki). Więc kiedy skorzystamy z jednostki procentowej w elemencie `children` obliczy on szerokość na podstawie szerokości okna. Chyba, że jeszcze jakiś przodek ma ustawioną szerokość.



## HTML

```
<div class="parent">
  Jestem rodzicem
  <div class="child">
    Jestem dzieckiem
  </div>
</div>
```

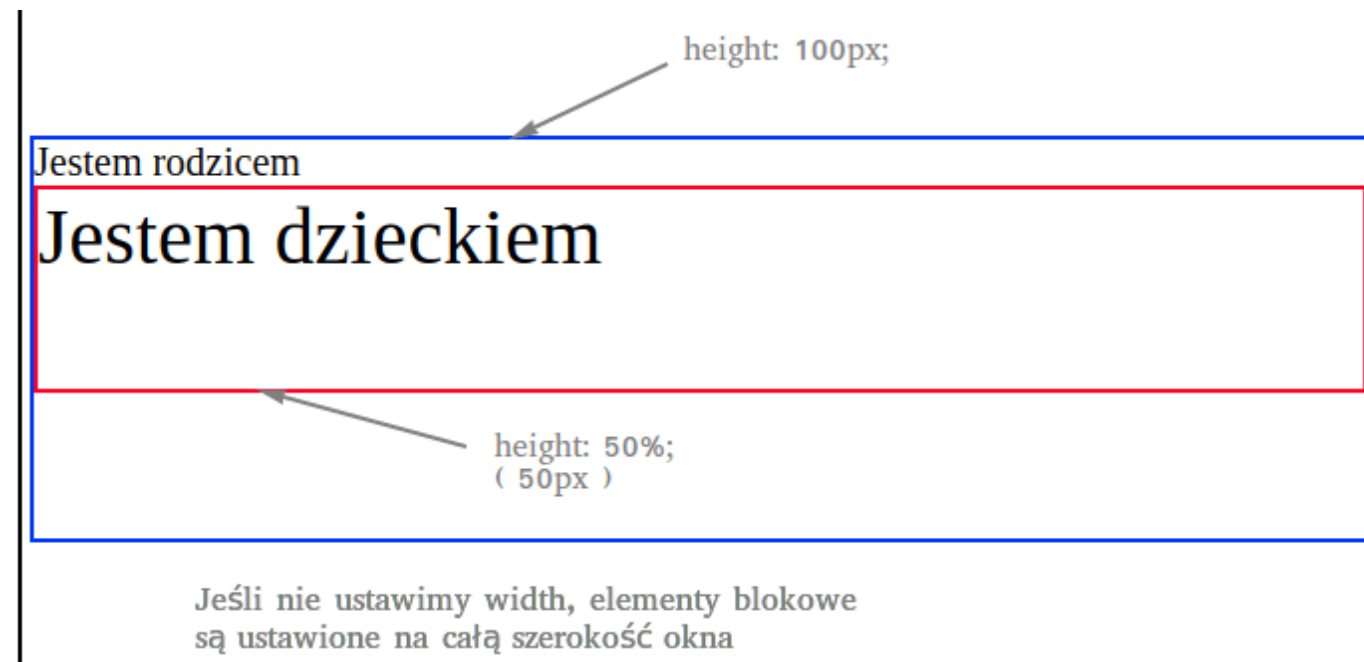
## CSS

```
.parent {
  font-size: 10px;
  border: 1px solid blue;
}
.child {
  font-size: 200%;
  width: 50%;
  border: 1px solid red;
}
```

# Procent – height

## Określenie height w procentach

Jeśli chodzi o własność `height` to tutaj bardzo ważne jest, aby najbliższy rodzic miał ustawioną konkretną wysokość.



## HTML

```
<div class="parent">
  Jestem rodzicem
  <div class="child">
    Jestem dzieckiem
  </div>
</div>
```

## CSS

```
.parent {
  font-size: 10px;
  height: 100px;
  border: 1px solid blue;
}
.child {
  font-size: 200%;
  height: 50%;
  border: 1px solid red;
}
```

# Procent – height

## Określenie height w procentach

W przypadku jeśli nie ustawimy dla elementu parent wysokości, element children nie będzie wiedział o jaki procent nam chodzi.



## HTML

```
<div class="parent">
  Jestem rodzicem
  <div class="child">
    Jestem dzieckiem
  </div>
</div>
```

## CSS

```
.parent {
  font-size: 10px;
  border: 1px solid blue;
}
.child {
  font-size: 200%;
  height: 50%;
  border: 1px solid red;
}
```

# Procent – height - ważne

## Określenie height w procentach

Ważne, żeby sobie zdać sprawę z tego, że w przypadku ustawiania height, brany jest pod uwagę **najbliższy rodzic** - jeśli ustawisz height dla przodka, który nie jest najbliższym rodzicem, wysokość nie będzie ustawiona.



## HTML

```
<div class="parent">
  Jestem rodzicem
  <div class="child">
    Jestem dzieckiem
  </div>
</div>
```

## CSS

```
html, body {
  height: 100%;
}
.parent {
  font-size: 10px;
  border: 1px solid blue;
}
.child {
  font-size: 200%;
  height: 50%;
  border: 1px solid red;
}
```

# Jednostki viewportu a font-size

## Responsywne czcionki

Jednostki viewportu można użyć do ustawienia wielkości czcionki.

Rozmiar tekstu będzie się zmieniał w zależności od wielkości obszaru roboczego (viewportu).

## CSS

```
h2 {  
    font-size: 6vw;  
}  
h3 {  
    font-size: 2vw;  
}
```

# Jednostki viewportu – inne wielkości

## Jednostki relatywne do viewportu

**1vw** = 1% szerokości viewportu;

**1vh** = 1% wysokości viewportu;

**1vmin** = **1vw** lub **1vh**, która wartość jest mniejsza;

**1vmax** = **1vw** lub **1vh**, która wartość jest większa;

Z tych własności wynika, że w przypadku viewportu (tzw. obszaru roboczego) ekranu o szerokości 1000px wartość dla 1vw jest równa:  $1\% \times 1000\text{px} = 10\text{px}$ .

## HTML

```
<div class="parent"></div>
```

## CSS

```
.parent {  
  height: 50vh;  
  width: 50vw;  
  border: 1px solid blue;  
}
```

Element będzie zajmował 50% wysokości i szerokości okna.

# vh vs %

## Jednostki relatywne do viewportu

Jeśli chcielibyśmy ustawić element na wysokość 100% w następujący sposób:

```
.parent {  
  height: 100%;  
}
```

...to doskonale wiesz, że to nie zadziała, dopóki nie ustawimy wysokości dla `html` oraz `body`. Ale element `parent` nie może mieć już żadnego innego przodka:

```
html, body {  
  height: 100%;  
}
```

Można użyć w takiej sytuacji `vh` i nie musimy się martwić o zagnieżdżenie ani ustawianie innym elementom jakieś wysokości.

```
.parent {  
  height: 100vh;  
}
```