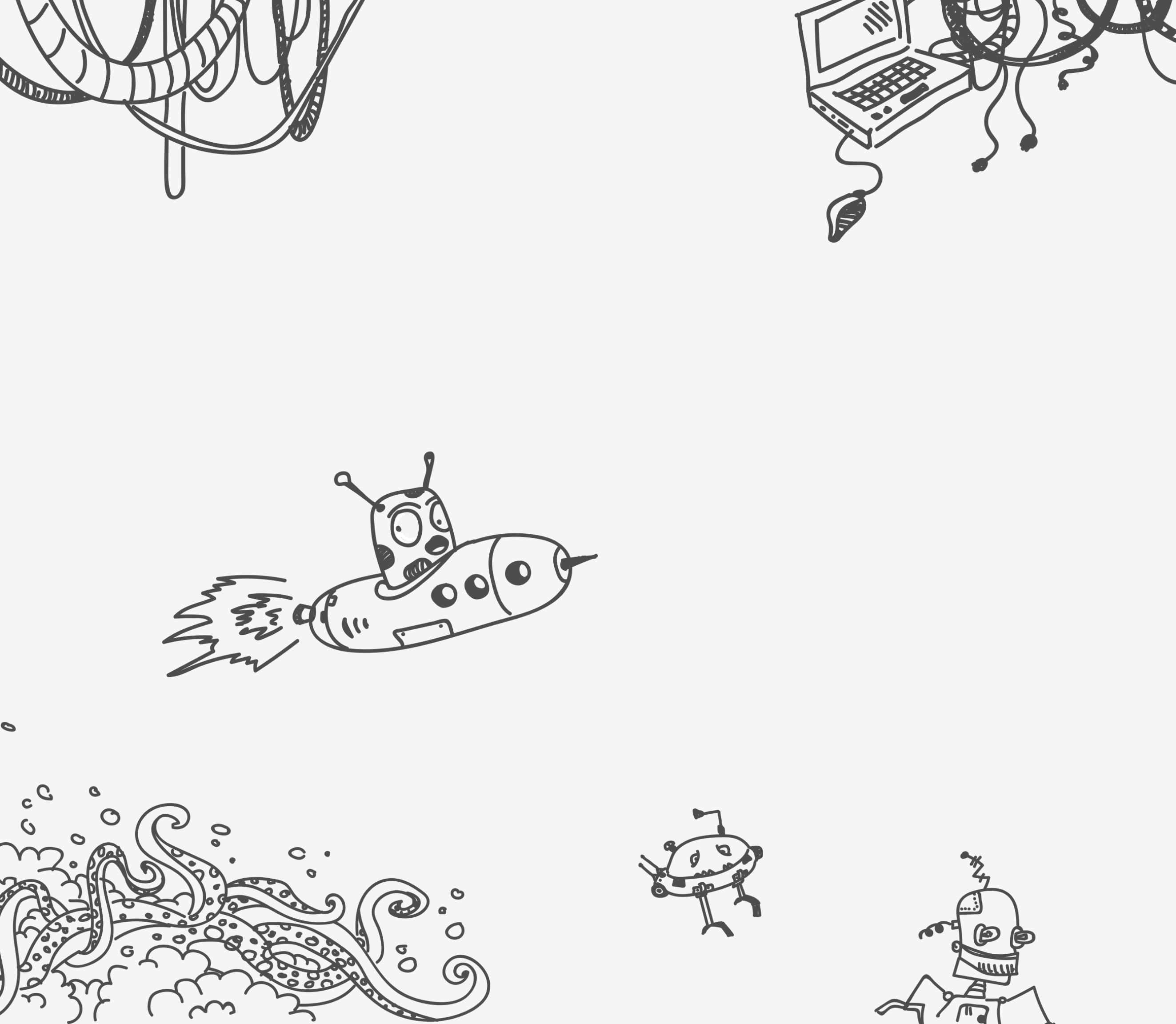


Zaawansowane selektory



Pseudoklasy

Pseudoklasy

Chcesz ostylewać kliknięty link czy np. zaznaczony przycisk wyboru?

Chcesz nadać kolor np. ostatniemu wierszowi w liście?

W takich przypadkach najlepiej użyć **pseudoklas** i **pseudoelementów**.

Selektor i **pseudoklasę** oddzielamy dwukropkiem.

```
a:hover {  
  color: red;  
}  
  
a:visited {  
  color: violet;  
}
```

:hover

Dynamiczne pseudoklasy pozwalają wybrać elementy, z którymi następuje albo nastąpiła interakcja.

Efekt po najechaniu

Dzięki tej pseudoklasie znajdziesz element, nad którym znajduje się kursor myszy. Jest to bardzo przydatne m.in. podczas tworzenia animacji na stronie.



Nie poznaliśmy jeszcze właściwości `transition`. Służy ona do płynnego przejścia między dwoma stanami elementu. Więcej o niej będziemy mówić w kolejnych modułach.

Przykład

```
.element {  
  height: 100px;  
  background: orangered;  
  transition: 0.5s;  
}  
.element:hover {  
  background: springgreen;  
}
```

Efekt



Najedź myszką na element!

:active

Efekt po naciśnięciu

Dzięki tej pseudoklasie wybierzesz element, który jest właśnie naciskany.

Używane jest to np. do stworzenia efektu wciskanego guzika (przez zmianę jego cienia).

Przykład

```
.btn {  
  padding: 20px 60px;  
  color: #fff;  
  background: orangered;  
  transition: 0.2s;  
  cursor: pointer;  
}  
.btn:active {  
  box-shadow: inset 0 2px 3px rgba(0,0,0,0.5)  
  background: deepskyblue;  
}
```

Efekt



Kliknij w przycisk!

:visited

Efekt po odwiedzeniu linka

Oznacza linki prowadzące do stron już odwiedzonych. Linki muszą prowadzić do realnych stron.



Stylowanie pseudoklasy `:visited` ma bardzo duże ograniczenia - np. nie da się zmieniać koloru tła elementu gdy dany element nie ma początkowego tła.

Spowodowane jest to bezpieczeństwem. Gdyby takich obostrzeń nie było, moglibyśmy w prosty sposób czytać prywatne dane użytkownika (np. jakie strony odwiedził).

Przykład

```
a:visited {  
  color: #999;  
}
```

Efekt

<http://google.pl>
<http://wp.pl>
<http://onet.pl>
<https://tympanus.net/codrops/>
<https://www.webdesignerdepot.com/>
<https://grafmag.pl/>

Kliknij w jakiś link!

:focus

Efekt wyróżniający

Dzięki tej pseudoklasie jesteśmy w stanie znaleźć element który ma `focus` (jest w jakiś sposób wybrany).

Ta pseudoklasa jest najczęściej używana do wyróżnienia aktywnego elementu formularza (w którym aktualnie jest kursor) lub linku.



Elementy, które można obsługiwać za pomocą klawiatury (pola formularzy, linki, elementy z atrybutem `tabindex`) mają domyślną niebieską obwódkę. Można się jej pozbyć za pomocą właściwości `outline: none`.

Usuwanie tej obwódki **zawsze** trzeba dać jakieś alternatywne zaznaczenie wybranych elementów. W przeciwnym razie użytkownik nie będzie wiedział gdzie się znajduje na stronie poruszając się po niej za pomocą klawiatury. Jest to jedna z najgorszych rzeczy jakie można zrobić dla dostępności naszej strony.

Przykład

```
a:focus,  
button:focus,  
input:focus {  
    outline: none;  
    border: 2px dotted #333;  
}
```

Efekt

Link

Button

Input

Kliknij w elementy!

LoVe and HAte

Zasada kolejności

Przy stosowaniu pseudoklas dla linków należy trzymać się określonego porządku.

Prawidłowa kolejność użycia pseudoklas jest podana po prawej.

```
a:link { }  
a:visited { }  
a:hover { }  
a:active { }
```

Pseudoklasa **:link** odnosi się do nieodwiedzonego linku (elementu **a**) który posiada atrybut **href**.

Pseudoklasy dla formularzy

:enabled

Dzięki tej pseudoklasie jesteśmy w stanie znaleźć element, który jest włączony.

```
input[type="text"]:enabled { }  
input[type="text"]:disabled { }  
input[type="checkbox"]:checked { }
```

:disabled

Dzięki tej pseudoklasie znajdziemy element, który jest wyłączony.

:checked

Dzięki tej pseudoklasie znajdziemy checkboksy, które są wybrane.

:first-child, :last-child

:first-child

Pseudoklasa, która wybiera z danej kolekcji element, który jest pierwszym dzieckiem.

Przypuśćmy, że za pomocą selektora wybraliśmy wszystkie `li`. Jeżeli chcemy pobrać te li, które są pierwszym dzieckiem, wtedy użyjemy `li:first-child`

```
li {}  
li:first-child {}  
  
div span {}  
div span:first-child {}
```

:last-child

Pseudoklasa, która jest odwrotnością `:first-child`. Pobiera element, który jest ostatnim dzieckiem.

```
div {}  
div:last-child {}  
  
div strong {}  
div strong:last-child {}
```

:first-child, :last-child

:first-child

Pseudoklasa, która wybiera z danej kolekcji element, który jest pierwszym dzieckiem.

Przypuśćmy, że za pomocą selektora wybraliśmy wszystkie `li`. Jeżeli chcemy pobrać te `li`, które są pierwszym dzieckiem, wtedy użyjemy `li:first-child`

```
li {}  
li:first-child {}  
  
div span {}  
div span:first-child {}
```

Kolekcja wszystkich `li`

:last-child

Pseudoklasa, która jest odwrotnością `:first-child`. Pobiera element, który jest ostatnim dzieckiem.

```
div {}  
div:last-child {}  
  
div strong {}  
div strong:last-child {}
```

:first-child, :last-child

:first-child

Pseudoklasa, która wybiera z danej kolekcji element, który jest pierwszym dzieckiem.

Przypuśćmy, że za pomocą selektora wybraliśmy wszystkie `li`. Jeżeli chcemy pobrać te `li`, które są pierwszym dzieckiem, wtedy użyjemy `li:first-child`

```
li {}  
li:first-child {}  
  
div span {}  
div span:first-child {}
```

`li`, które są pierwszym dzieckiem

:last-child

Pseudoklasa, która jest odwrotnością `:first-child`. Pobiera element, który jest ostatnim dzieckiem.

```
div {}  
div:last-child {}  
  
div strong {}  
div strong:last-child {}
```

:first-child, :last-child

:first-child

Pseudoklasa, która wybiera z danej kolekcji element, który jest pierwszym dzieckiem.

Przypuśćmy, że za pomocą selektora wybraliśmy wszystkie `li`. Jeżeli chcemy pobrać te `li`, które są pierwszym dzieckiem, wtedy użyjemy `li:first-child`

```
li {}  
li:first-child {}
```

```
div span {}  
div span:first-child {}
```

Wszystkie spany w divach

:last-child

Pseudoklasa, która jest odwrotnością `:first-child`. Pobiera element, który jest ostatnim dzieckiem.

```
div {}  
div:last-child {}  
  
div strong {}  
div strong:last-child {}
```

:first-child, :last-child

:first-child

Pseudoklasa, która wybiera z danej kolekcji element, który jest pierwszym dzieckiem.

Przypuśćmy, że za pomocą selektora wybraliśmy wszystkie `li`. Jeżeli chcemy pobrać te li, które są pierwszym dzieckiem, wtedy użyjemy `li:first-child`

```
li {}  
li:first-child {}
```

```
div span {}  
div span:first-child {}
```

spany, które są pierwszym dzieckiem w divach

:last-child

Pseudoklasa, która jest odwrotnością `:first-child`. Pobiera element, który jest ostatnim dzieckiem.

```
div {}  
div:last-child {}  
  
div strong {}  
div strong:last-child {}
```

:first-child, :last-child

:first-child

Pseudoklasa, która wybiera z danej kolekcji element, który jest pierwszym dzieckiem.

Przypuśćmy, że za pomocą selektora wybraliśmy wszystkie `li`. Jeżeli chcemy pobrać te li, które są pierwszym dzieckiem, wtedy użyjemy `li:first-child`

```
li {}  
li:first-child {}  
  
div span {}  
div span:first-child {}
```

:last-child

Pseudoklasa, która jest odwrotnością `:first-child`. Pobiera element, który jest ostatnim dzieckiem.

```
div {}  
div:last-child {}  
  
div strong {}  
div strong:last-child {}
```

Kolekcja wszystkich `div`

:first-child, :last-child

:first-child

Pseudoklasa, która wybiera z danej kolekcji element, który jest pierwszym dzieckiem.

Przypuśćmy, że za pomocą selektora wybraliśmy wszystkie `li`. Jeżeli chcemy pobrać te `li`, które są pierwszym dzieckiem, wtedy użyjemy `li:first-child`

```
li {}  
li:first-child {}  
  
div span {}  
div span:first-child {}
```

:last-child

Pseudoklasa, która jest odwrotnością `:first-child`. Pobiera element, który jest ostatnim dzieckiem.

```
div {}  
div:last-child {}  
  
div strong {}  
div strong:last-child {}
```

`div`, które są ostatnim dzieckiem

:first-child, :last-child

:first-child

Pseudoklasa, która wybiera z danej kolekcji element, który jest pierwszym dzieckiem.

Przypuśćmy, że za pomocą selektora wybraliśmy wszystkie `li`. Jeżeli chcemy pobrać te li, które są pierwszym dzieckiem, wtedy użyjemy `li:first-child`

```
li {}  
li:first-child {}  
  
div span {}  
div span:first-child {}
```

:last-child

Pseudoklasa, która jest odwrotnością `:first-child`. Pobiera element, który jest ostatnim dzieckiem.

```
div {}  
div:last-child {}  
  
div strong {}  
div strong:last-child {}
```

Wszystkie strongi w div

:first-child, :last-child

:first-child

Pseudoklasa, która wybiera z danej kolekcji element, który jest pierwszym dzieckiem.

Przypuśćmy, że za pomocą selektora wybraliśmy wszystkie `li`. Jeżeli chcemy pobrać te `li`, które są pierwszym dzieckiem, wtedy użyjemy `li:first-child`

```
li {}  
li:first-child {}  
  
div span {}  
div span:first-child {}
```

:last-child

Pseudoklasa, która jest odwrotnością `:first-child`. Pobiera element, który jest ostatnim dzieckiem.

```
div {}  
div:last-child {}  
  
div strong {}  
div strong:last-child {}
```

`strong`, które są ostatnim dzieckiem w `div`ach

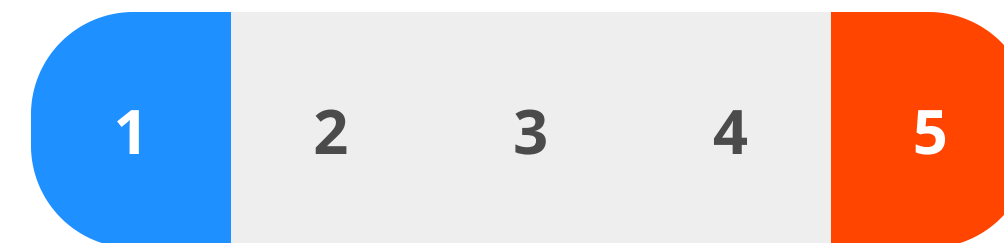
:first-child, :last-child

Przykład

```
ul {  
  display: flex;  
  list-style: none;  
  padding: 0;  
  font-weight: 700;  
}  
li {  
  padding: 40px;  
  background: #eee;  
}  
li:first-child {  
  background: dodgerblue;  
  color: white;  
  border-radius: 50px 0 0 50px;  
}  
li:last-child {  
  background: orangered;  
  color: white;  
  border-radius: 0 50px 50px 0;  
}
```

```
<ul>  
  <li>1</li>  
  <li>2</li>  
  <li>3</li>  
  <li>4</li>  
  <li>5</li>  
</ul>
```

Efekt



:nth-child()

Pseudoklasa, która wybiera n-ty element potomny swojego rodzica.

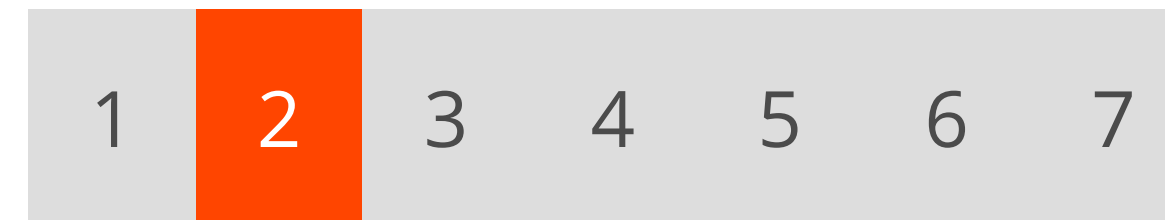
W nawiasy wpisujemy formułę `a n + b`, gdzie:

- `a` – liczba całkowita
- `n` – określenie wielokrotności (jak w pętli)
- `b` – liczba całkowita

Przykład

```
li:nth-child(2) {  
  background: orangered;  
  color: #fff;  
}
```

Efekt



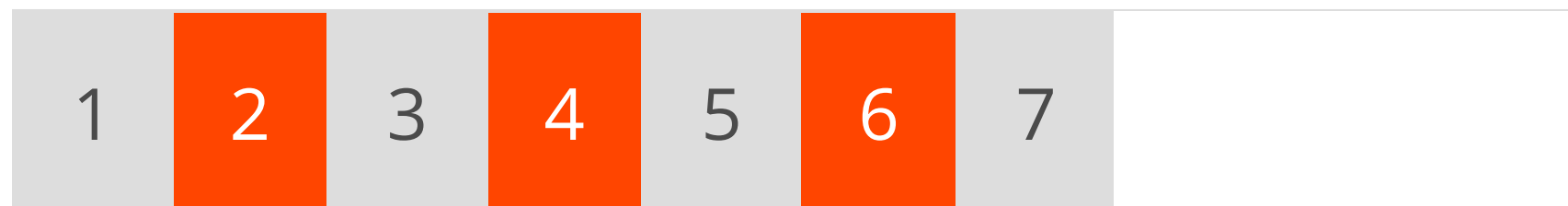
Pseudoklasy związane z położeniem

Wybieramy `li`, które jest co drugie, zaczynając od 2.

Przykład

```
li:nth-child(2n) {  
  background: orangered;  
  color: #fff;  
}
```

Efekt



Wybieramy `li`, które są co 2 dzieckiem zaczynając od 5.

Przykład

```
li:nth-child(2n+5) {  
  background: orangered;  
  color: #fff;  
}
```

Efekt



:nth-child

n	$2n+1$	$4n+1$	$4n+4$	$4n$	$5n+2$
0	1	1	4	0	2
1	3	5	8	4	7
2	5	9	12	8	12
3	7	13	16	12	17
4	9	17	20	16	22

Na poniższych stronach możesz potrenować selektory nth:

- <http://css-tricks.com/examples/nth-child-tester>
- <http://nth-test.com/>

:nth-last-child()

Pseudoklasa, która wybiera element, który jest n-tym dzieckiem swojego rodzica **licząc od końca**.

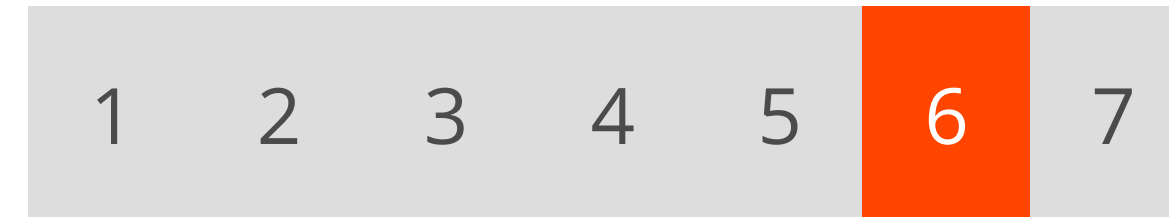
W nawiasy wpisujemy formułę `a n + b`, gdzie:

- `a` – liczba całkowita
- `n` – określenie wielokrotności (jak w pętli)
- `b` – liczba całkowita

Przykład

```
li:nth-last-child(2) {  
  background: orangered;  
  color: #fff;  
}
```

Efekt



:first-of-type

Pseudoklasa, która wybiera pierwszy element danego typu.

Przykład

```
strong:first-of-type {  
  color: red;  
  font-weight: bold;  
}
```

```
<div>  
  <span>1</span>  
  <strong>2</strong>  
  <strong>3</strong>  
  <span>4</span>  
</div>
```

Efekt

1 2 3 4

:last-of-type

Pseudoklasa, która wybiera ostatni element danego typu.

Przykład

```
strong:last-of-type {  
  color: red;  
  font-weight: bold;  
}
```

```
<div>  
  <span>1</span>  
  <strong>2</strong>  
  <strong>3</strong>  
  <span>4</span>  
</div>
```

Efekt

1 2 **3** 4

:nth-of-type

Pseudoklasa `nth-of-type` działa bardzo podobnie do `nth-child`, z tym że, wybiera element, który jest n-ty danego typu (danego znacznika). Nie ma możliwości pobrania "nty-danej-klasy":

Poniższy przykład pokazuje różnicę:

```
div strong:nth-child(2) {  
  background: #F17F42;  
  color: #fff;  
  font-weight: bold;  
}
```

```
<div>  
  <span>1</span>  
  <strong>2</strong>  
  <strong>3</strong>  
</div>
```

1

2

3

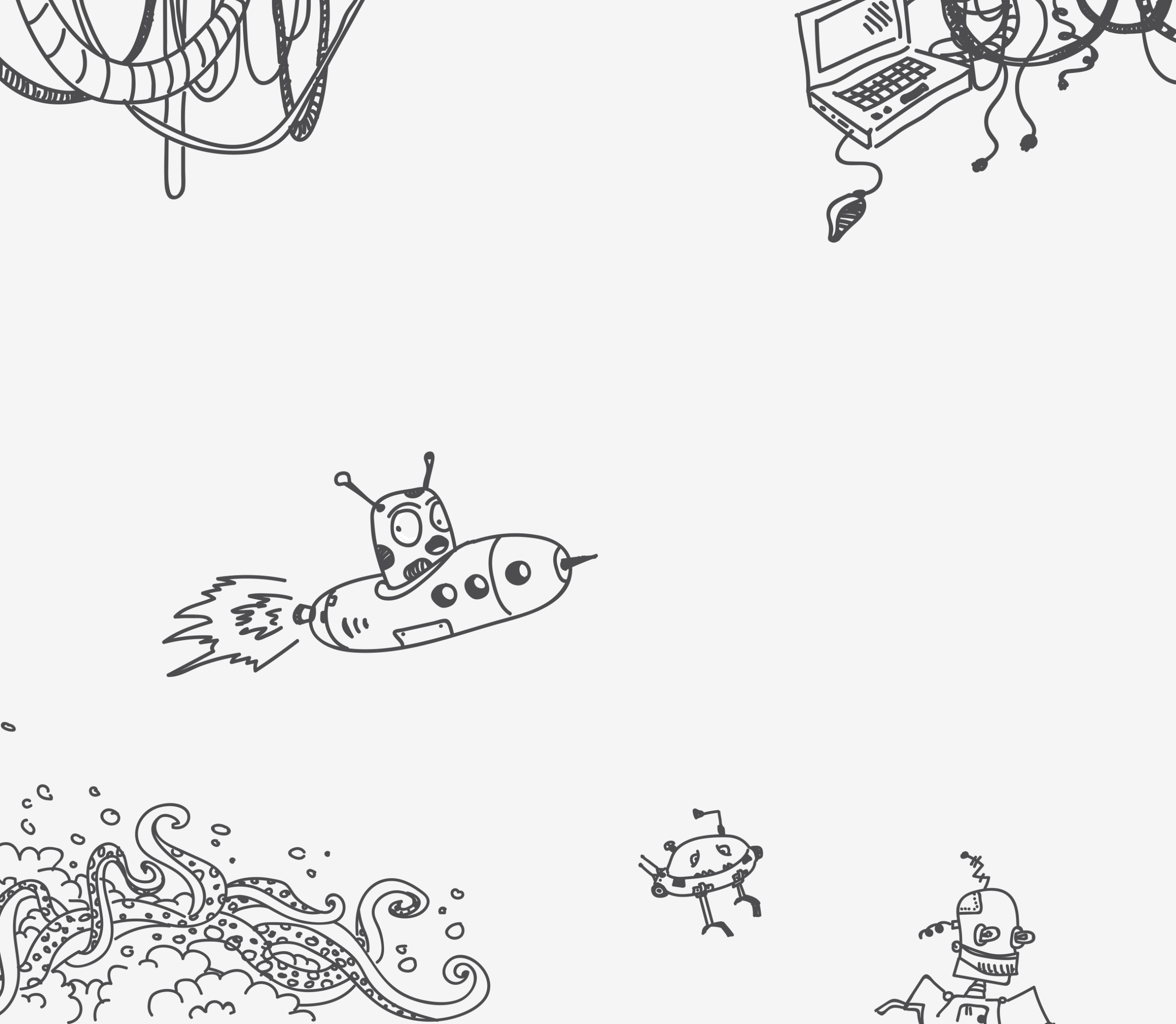
```
div strong:nth-of-type(2) {  
  background: #F17F42;  
  color: #fff;  
  font-weight: bold;  
}
```

```
<div>  
  <span>1</span>  
  <strong>2</strong>  
  <strong>3</strong>  
</div>
```

1

2

3



Pseudoelementy

Pseudoelementy

Czym są pseudoelementy?

Poza pseudoklasami w CSS mamy też pseudoelementy.

Są one używane do stylowania części danego elementu albo treści wokół elementu. Możemy też stworzyć element HTML przy pomocy CSS. Pseudoelementy są nieklikalne.



Pojedynczy a podwójny dwukropek

Dla pseudoelementów można spotkać zapis z podwójnym i pojedynczym dwukropkiem. Wynika to z tego, że początkowo powyższe pseudoelementy były zapisywane z pojedynczym dwukropkiem.

W późniejszym czasie twórcy CSS wprowadzili podwójny zapis by oddzielić pseudoelementy od pseudoklas (np. :hover).

::before

Pseudoelement używany do dodania danej treści jako pierwsze dziecko danego elementu.

::after

Pseudoelement używany do dodania danej treści jako ostatnie dziecko danego elementu.

```
p::before {  
}  
  
ul::after {  
}
```

Atrybut content

Podczas używania pseudoelementów **wymagany jest** atrybut `content`. Atrybut ten działa tylko dla pseudoelementów.

Przykład

```
ul::before {  
  content: "List start";  
  color: blue;  
  font-weight: bold;  
}
```

```
ul::after {  
  content: "List end";  
  color: red;  
  font-weight: bold;  
}
```

```
<ul>  
  <li>1</li>  
  <li>2</li>  
</ul>
```

Efekt

List start

- 1
- 2

List end

::before i ::after w praktyce

```
.title {
  display: flex;
  justify-content: center;
  align-items: center;
}

.title::before,
.title::after {
  content: "";
  display: block;
  height: 3px;
  background: #222;
  flex-grow: 1;
}

.title strong {
  color: #fff;
  display: inline-block;
  padding: 10px 30px;
  color: #222;
}
```

```
<h3 class="title">
  <strong>Tekst tytułu</strong>
</h3>
```

Efekt

————— **Tekst tytułu** —————



Używamy w tym przykładzie `flex` w celu ustawienia elementów obok siebie. Będziemy o nim mówić podczas tego modułu.

Pseudoelementy związane z tekstem

::first-letter

Dzięki temu pseudoelementowi wystylujemy pierwszą **literę** tekstu.

Przykład

```
.element::first-letter {  
  color: #fff;  
  background: #2EC4B6;  
}
```

Efekt

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aperiam eveniet exercitationem inventore iure laboriosam modi nostrum, perferendis placeat saepe sapiente!

::first-line

Dzięki temu pseudoelementowi wystylujemy pierwszą **linię** tekstu.

Przykład

```
.element::first-line {  
  color: #fff;  
  background: #2EC4B6;  
}
```

Efekt

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aperiam eveniet exercitationem inventore iure laboriosam modi nostrum, perferendis placeat saepe sapiente!

Zaprzeczenie w CSS

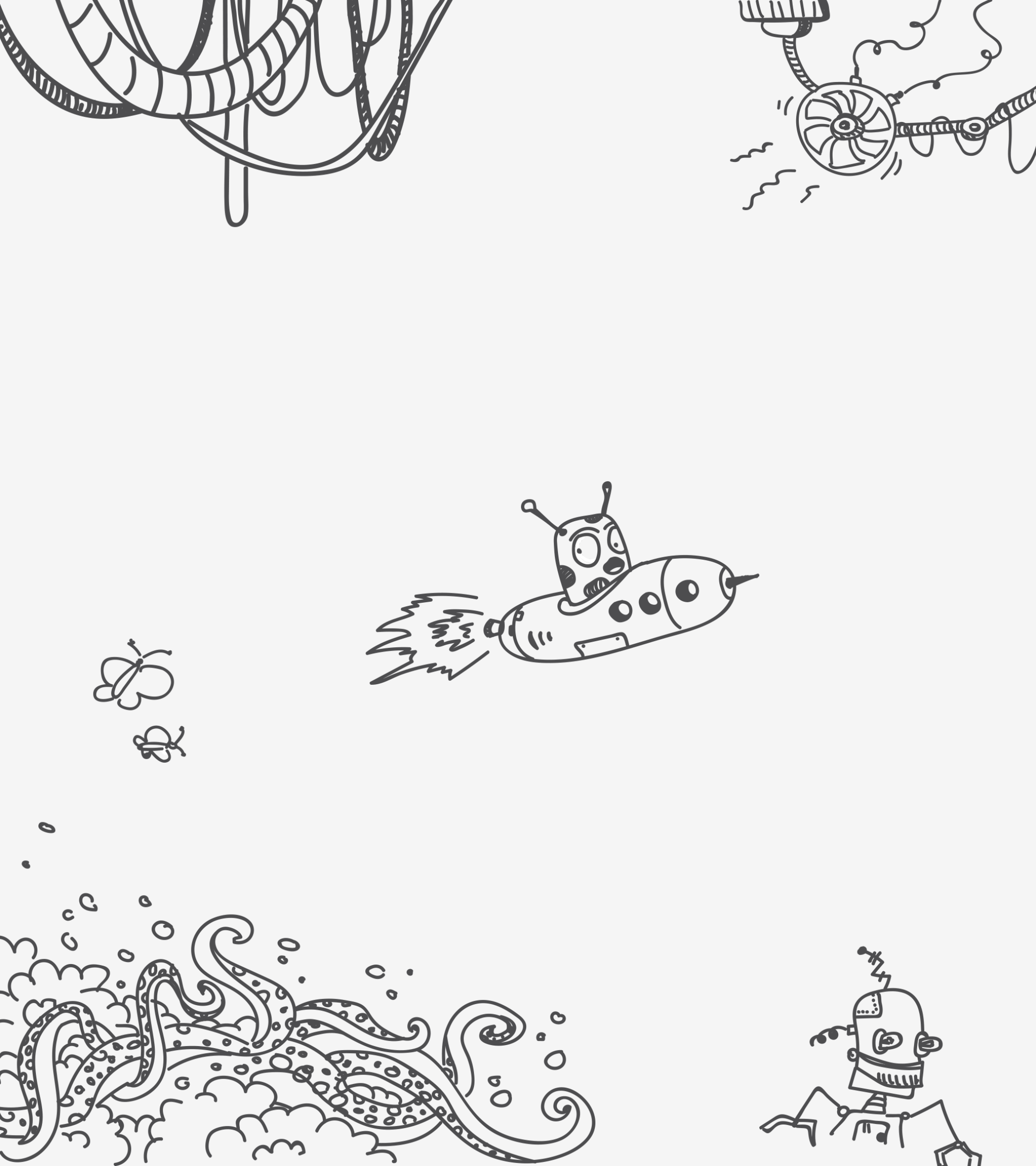
Jeżeli nie chcemy czegoś wybrać?

CSS pozwala nam na używanie zaprzeczeń dzięki selektorowi `:not`.

W pierwszym przykładzie wybieramy wszystkie elementy `div` oprócz elementów o klasie `content`.

W drugim przykładzie odwrotnie – wszystkie elementy o klasie `content`, ale nie elementy `div`.

```
div:not(.content) {  
  color: #F00;  
}  
.content:not(div) {  
  color: #F00;  
}
```

Spis selektorów, które warto pamiętać.