

Niedystynktywne stylizowanie inputów

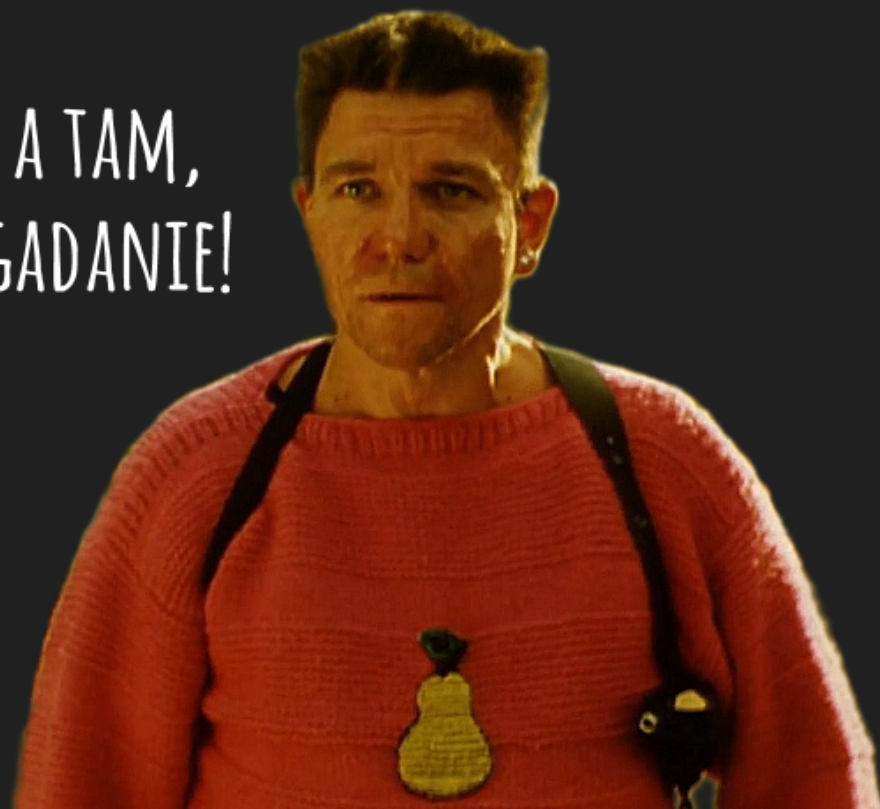
używamy selektora input i część

```
input
{
  width: 300px;
  background-color: #efefef;
  color: #666;
  border: 2px solid #ddd;
  border-radius: 5px;
  font-size: 20px;
  padding: 10px;
  box-sizing: border-box;
  outline: none;
  margin-top: 10px;
}
```

wada: niejako "przy okazji" ostylujemy wszystkie inputy, m.in. text, password, submit, button, file, color, date, email, month, number, time, week itd.

Nie stosuj! No chyba że uważasz, iż ignorancja to błogosławieństwo

A TAM,
GADANIE!



Dystynktywne stylizowanie inputów

konkretnie określamy, które
inputy chcemy ostylować

używamy zapisu w postaci:
`input[type=typ_pola]`

```
input[type=text],  
input[type=password]  
{  
    width: 300px;  
    background-color: #efefef;  
    color: #666;  
    border: 2px solid #ddd;  
    border-radius: 5px;  
    font-size: 20px;  
    padding: 10px;  
    box-sizing: border-box;  
    outline: none;  
    margin-top: 10px;  
}
```

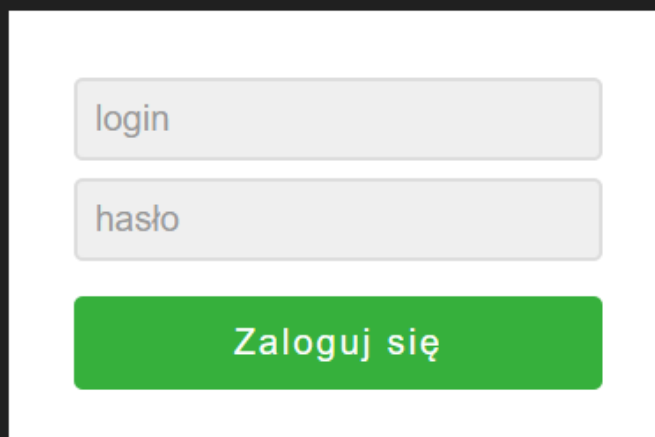
OD STOSOWANIA
INNYCH ZAPISÓW
MASZ ŚMIECH JAK
ŚWINIA KASZEL



Atrybut HTML placeholder

placeholder = tekst lub symbol zastępczy, tymczasowo umieszczony w polu edycyjnym (do czasu wpisania "prawdziwej" zawartości pola).

```
<input type="text" placeholder="login">  
<input type="password" placeholder="hasło">
```



A login form with two input fields and a button. The first input field is labeled 'login' and the second is labeled 'hasło'. Below the input fields is a green button labeled 'Zaloguj się'.

do czasu wpisania
loginu lub hasła
placeholder "trzyma
miejsce" w polu

Zmiana koloru placeholdera

```
input::-webkit-input-placeholder
{
    color: #999;
}
```

```
input:focus::-webkit-input-placeholder
{
    color: #428C42;
}
```

```
/* Firefox < 19 */
input:-moz-placeholder
{
    color: #999;
}
```

```
input:focus:-moz-placeholder
{
    color: #428C42;
}
```

```
/* Firefox > 19 */
input::-moz-placeholder
{
    color: #999;
}
```

```
input:focus::-moz-placeholder
{
    color: #428C42;
}
```

```
/* Internet Explorer 10 */
input:-ms-input-placeholder
{
    color: #999;
}
```

```
input:focus:-ms-input-placeholder
{
    color: #428C42;
}
```

Ostylowanie zyskania aktywności przez pole

używamy `:focus`, tak jak wcześniej używaliśmy `:hover`

```
input[type=text]:focus,
input[type=password]:focus {
  -webkit-box-shadow: 0px 0px 10px 2px rgba(204,204,204,0.9);
  -moz-box-shadow: 0px 0px 10px 2px rgba(204,204,204,0.9);
  box-shadow: 0px 0px 10px 2px rgba(204,204,204,0.9);
  border: 2px solid #a5cda5;
  background-color: #e9f3e9;
  color: #428c42;
}
```

Diagram illustrating the box-shadow syntax: `0px 0px 10px 2px rgba(204,204,204,0.9)`. The values are labeled as follows: `0px` (poziom), `0px` (pion), `10px` (radius), `2px` (spread), and `rgba(204,204,204,0.9)` (color + opacity). The `blur` property is also indicated.

Użycie "pudełkowego" cienia `box-shadow`

generator z odcinka znajdziesz pod adresem:

www.cssmatic.com/box-shadow

Nadużywanie br-ów

```
<div id="container">
  <form>

    Login
    <input type="text">

    <br /><br />

    Hasło
    <input type="password">

    <br /><br />

    <input type="submit"
      value="Zaloguj się">

  </form>
</div>
```

Wady użycia br-ów
jako odstępów pionowych:

1. Wysokość dwóch br-ów będzie różna w zależności od przeglądarki internetowej
2. Określamy wygląd strony poza warstwą CSS (w HTML).

Domyślne sposoby naliczania rozmiaru

box-sizing: content-box;

domyślny sposób dla pól text, password - szerokość obliczana jest jako suma szerokości zawartości (contentu) oraz paddingu

box-sizing: border-box;

domyślny sposób dla inputa submit (przycisku) - szerokość definiowana jest w CSS i jest stała, co sprawia, że padding jest odejmowany od dostępnej, określonej w CSS przestrzeni

Prefixy niektórych właściwości CSS

- webkit** = silnik renderowania stron użyty w przeglądarkach Safari, Chrome, niektórych wersjach Opery
- moz** = Mozilla Firefox
- ms** = Internet Explorer
- o** = Opera

Takie przedrostki stosujemy aby uniknąć pewnych niekonsekwencji w implementacji sposobu renderowania CSS w przeróżnych przeglądarkach (w końcu tworzą je osobne zespoły ludzi). Takie prefixy zostaną z czasem usunięte (a przynajmniej w teorii), kiedy dana właściwość doczeka się oficjalnego sposobu implementacji w przeglądarkach.

Usunięcie placeholdera przy zyskaniu aktywności

this - wskaźnik (uchwyt, referencja) obiektu, na rzecz którego dana metoda (tutaj obsługa zdarzenia) została wywołana.

```
<input type="text" placeholder="login" onfocus="this.placeholder=''"  
onblur="this.placeholder='login'" >
```

używamy apostrofów, ponieważ cała wartość atrybutu jest zapisana w cudzysłowie

```
<input type="password" placeholder="hasło" onfocus="this.placeholder=''"  
onblur="this.placeholder='hasło'" >
```

zdarzenie utraty aktywności przez obiekt

zdarzenie, które zachodzi w momencie zyskania aktywności przez obiekt - u nas pole edycyjne