

Tworzenie Aplikacji Internetowych

Laboratorium 2

Style CSS

Style CSS służą do definicji sposobu wyświetlania elementów na stronie HTML. Można je umieścić:

- w zewnętrznym pliku .css podając w dokumencie HTML źródło do pliku w znaczniku **head**:
<link rel="stylesheet" href="plik.css">
- wewnątrz znacznika **style** umieszczonego w znaczniku **head**
- wewnątrz znaczników dla których chcemy zmienić style (zmieniają tylko styl danego znacznika) poprzez atrybut style: <znacznik style="...">

Style CSS definiuje się w następujący sposób:

```
selektor {  
    właściwość: wartość;  
    właściwość: wartość;  
    ...  
}
```

Selektor może odwoływać się do:

- **znacznika HTML** (styl zostaje zastosowany do wszystkich takich znaczników w dokumencie HTML, w przypadku zmiany stylu wyświetlania tekstu zostaje on dodatkowo zastosowany do wszystkich znaczników wewnątrz)
- obiektów o przypisanych atrybutach **klasy** (każdy znacznik, którego styl chcemy zmienić musi posiadać przypisany **atrybut** class="nazwaklasy", a selektor należy zapisać w formie **.nazwaklasy**)
- obiektu o przypisanym konkretnym identyfikatorze (dany znacznik musi posiadać **atrybut** określający jego identyfikator: id="nazwaidentyfikatora", a selektor należy zapisać w formie **#identyfikator**)

Przykład:

```
<head>  
  <meta charset="utf-8">  
  <style>  
    p { color: royalblue; } /* zmiana stylu wszystkich znaczników p */  
    .wazne { background: red; } /* zmiana stylu wszystkich znaczników z class="wazne" */  
    #haslo { color: gray; } /* zmiana stylu znacznika z id="haslo" */  
  </style>  
</head>
```

Na jeden element może wpływać **wiele stylów**. Jeżeli dla elementu są podane nowe właściwości – dopełniają one dany styl (akumulują się), jeżeli właściwości są zdublowane, to styl stosowany jest według kaskadowej hierarchii:

- Style z danego **znacznika** → Style z sekcji **head** i znacznika **style** → Style z zewnętrznego **pliku**
- Style z danego **znacznika** → Style z atrybutu **id** → Style z atrybutu **class** → Style **znacznika** → Style znacznika **zewnętrznego** → ...

```
<head>  
  <meta charset="utf-8">  
  <style>  
    div { color: royalblue; } /* kolor tekstu w div - niebieski */  
    p { color: red; } /* kolor tekstu w p - czerwony */  
    .akapit { color: gray; } /* kolor klasy 'akapit' - szary */  
  </style>
```

```
</head>
<body>
  <div class="header">
    <p class="akapit">Test</p> <!-- kolor tekstu będzie szary - class > znaczniki -->
  </div>
</body>
```

Wybrane właściwości:

Kolory:

- **tekstu:**
color: blue; /* wybór koloru ze zdefiniowanych */
color: #0000ff; /* kolor w postaci #RRGGBB */
color: #00f; /* kolor w postaci #RGB */
color: rgb(0, 0, 255);
- **tła:**
background: orange;
background: url("/sciezka/obrazek.png"); /* obrazek jako tło */
background: url("/sciezka/obrazek.png") **no-repeat**; /* nie powtarzaj */
background-position: right top; /* pozycja tła prawa góra – gdy no-repeat */
background-position: center; /* pozycja tła wyśrodkowana – gdy no-repeat */
background-size: cover; /* rozciągnij tło do wielkości elementu */

Marginesy i obramowania:

- **margines wewnętrzny** (kolor tła przestrzeni marginesu odpowiada kolorowi znacznika):
padding: 10px; /* 10 pikseli z każdej strony */
padding: 10px 0px 10px 20px; /* górny, prawy, dolny, lewy */
- **obramowanie wewnętrzne:**
border: 2px solid blue; /* obramowanie o górbkości 2 piksele, linia ciągła, niebieska */
border-style: solid; /* typ linii: dashed – przerywana, dotted – kropkowana, double – podwójna... */
border-width: 5px; /* grubość obramowania */
border-color: green; /* kolor obramowania */
border-radius: 10px; /* zaokrąglenie rogów */
- **margines zewnętrzny** (kolor tła przestrzeni odpowiada kolorowi tła elementów zewnętrznych):
margin: 10px; /* 10 pikseli z każdej strony */
margin: 10px 0px 10px 20px; /* górny, prawy, dolny, lewy */

Wysokość i szerokość:

- **szerokość:**
width: 500px;
width: 20%;
- **wysokość:**
height: 200px;
- **maksymalna szerokość/wysokość** (element zmienia wielkość do ustalonej):
max-width: 300px;
max-height: 200px;
- **minimalna szerokość/wysokość** (element się zwiększa od ustalonej):
min-width: 300px;
min-height: 200px;

Sposób wyświetlania znacznika:

- display: block; /* znacznik zajmuje całą szerokość strony */
- display: inline; /* znacznik zajmuje minimalną potrzebną przestrzeń */
- display: inline-block; /* znacznik zajmuje przestrzeń określoną np. przez właściwość width */
- display: none; /* znacznik jest ukryty – można go wyświetlić np. za pomocą JavaScript */

Czcionka i tekst:

- **Ustawienie położenia tekstu wewnątrz elementu:**
text-align: left; /* ustawienie z lewej strony */
text-align: center; /* wyśrodkowanie */
text-align: right; /* ustawienie z prawej strony */
text-align: justify; /* wyjustowanie */
- **Wyłączenie możliwości zaznaczania tekstu:**
user-select: none;
- **Dekoracja tekstu:**
text-decoration: none;
text-decoration: underline; /* podkreślenie */
text-decoration: line-through; /* przekreślenie */
text-decoration: overline; /* nadkreślenie */
- **Wielkość liter:**
text-transform: lower; /* tylko małe litery */
text-transform: upper; /* tylko duże litery */
text-transform: capitalize; /* pierwsza litera wyr. duża */
- **Wyświetlanie tekstu:**
text-indent: 10px; /* wcięcie tekstu */
letter-spacing: 2px; /* odstęp pomiędzy literami */
word-spacing: 10px; /* odstęp pomiędzy słowami */
line-height: 15px; /* wysokość linii */
text-shadow: 4px 4px 8px red; /* cień odstęp, odstęp, rozmycie */
- **Czcionka:**
font-family: "Rodzina", "Opcjonalna rodzina", ..., generyczna;
- **Przykładowe rodziny czcionek:**
Arial, "Times New Roman", Courier, "Lucida Console", ...
- **Czcionki generyczne:**
serif, sans-serif, monospace, cursive, fantasy
- **Rozmiar czcionki:**
font-size: 20px;
- **Styl czcionki:**
font-style: normal / italic / oblique;
- **Grubość czcionki:**
font-weight: 100 / 200 / 300 / normal / 500 / 600 / bold / 800 / 900

Przykład (uzupełnianie się styli nadanych w różny sposób – warto zwrócić uwagę na komentarze CSS i HTML):

```
<head>
  <meta charset="utf-8">
  <style>
    p { color: royalblue; background: seashell; } /* to dotyczy znaczników p */
    body { text-decoration: overline; } /* to dotyczy tekstu na całej stronie */
  </style>
</head>
<body>
  <p style="font-size:20px;font-family:Impact, 'Arial Narrow Bold', sans-serif">Tekst</p>
  <!-- powyższa zmiana przez style dotyczy tylko powyższego znacznika -->
</body>
```

Wynik:



Zadanie 1

W celu realizacji zadania należy:

- Utworzyć listę **ul** lub **ol** zawierającą **sześć** dowolnych elementów **li** zawierających dowolny tekst
- Zmienić **kolor tła** oraz **margines wewnętrzny** elementów **li** – zaobserwować zmiany
- Za pomocą różnego odwołania się do elementów listy (**class**, **id**, **style**) zmienić **kolor tekstu** poszczególnych elementów listy (nie zmieniając poprzednich ustawień dla znacznika **li**)
- Zmienić sposób wyświetlania elementów **li** na inline-block, ustawić **szerokość** (np. 150px), ustawić **margines zewnętrzny** i zablokować możliwość zaznaczania tekstu.
- **Wyśrodkować** tekst wewnątrz elementów **li** i zmienić **czcionkę** wszystkich elementów **li** (na wybraną).
- Zmienić **kolor tła** całej listy (**ul** lub **ol** – w zależności od użytej listy) i dodać jej **margines wewnętrzny**.
- Zaobserwować zachowanie się listy – również w przypadku zmniejszenia szerokości okna przeglądarki.

Zaawansowane modyfikacje selektorów i tranzycje (przejścia)

- **Pseudo-klasy selektorów** (dodaje się je po selektorach – np. div:hover):
:hover /* styl dla elementu po najechaniu kursorem */
:link /* styl dla nieodwiedzonego hiperłącza */
:visited /* styl dla odwiedzonego hiperłącza */
:active /* styl właśnie klikniętego hiperłącza/elementu */
:focus /* styl obecnie aktywnego elementu (np. input)
:checked /* styl dla zaznaczonego obiektu input typu radio/checkbox */
:nth-child(even) /* styl dla parzystego elementu zgodnego z znacznikiem (np. **tr** w tabeli)
:nth-child(odd) /* styl dla nieparzystego elementu zgodnego z znacznikiem (np. **tr** w tabeli)
:nth-child(n) /* styl dla n-tego elementu z kolejki (np. **tr** w tabeli)
:last-child /* styl ostatniego znacznika danego typu
:first-child /* styl pierwszego znacznika danego typu
::after /* dodaje element po obecnym */
::before /* dodaje element przed obecnym */
- **Oddzielanie selektorów** (znak przecinka):
selektorA, selektorB { ... } /* nadanie styli przez kilka selektorów jednocześnie */
- **Nadanie styli konkretnym podelementom selektora** (spacja)
selektorA selektorB { ... } /* styl zostanie nadany selektoromB wewnątrz selektorówA
- **Nadanie styli tylko bezpośrednim znacznikom wewnątrz selektora:**
selektorA > selektorB

Przykład:

```
<head>
  <meta charset="utf-8">
  <style>
    p, a { color: gray; } /* nadanie znacznikom p oraz a koloru tekstu */
    p { background: red; } /* nadanie koloru tła znacznika p */
    a:hover { color: green; } /* po najechaniu kursorem na znacznik nastąpi zmiana koloru tekstu */
    div p { background: yellow; } /* zmieni kolor tła znacznikom p umieszczonym wewnątrz div */
    div p:last-child { background: silver; } /* zmieni kolor tła ostatniego znacznika p z div */
  </style>
</head>
<body>
  <p>Tekst</p>
  <a href="#">LINK</a>
  <div>
    <p>Linia1</p><p>Linia2</p><p>Linia3 (ostatnia)</p>
  </div>
</body>
```

Wynik:

Tekst

LINK

Linia1

Linia2

Linia3 (ostatnia)

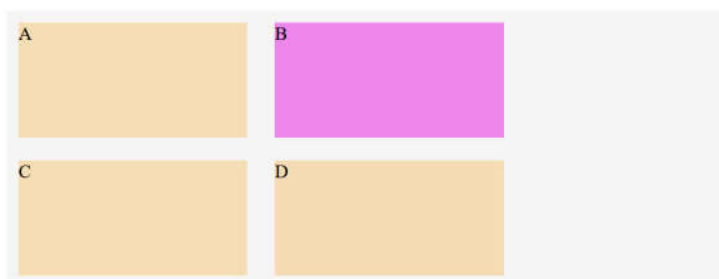
Przejścia pozwalają na płynną zmianę atrybutów (proste animacje)

- **Elementowi (bazowemu) należy dodać atrybut:**
transition: czas;
transition: właściwość czas; /* jeżeli animacja ma dotyczyć tylko danego atrybutu */
transition: właściwość1 czas1; właściwość2 czas2; /* różne czasy dla różnych atrybutów */
transition-delay: opóźnienie; /* opóźnienie */
- **Następnie należy stworzyć sytuację, w której styl zmieni swoją właściwość, np.:**
 - nadać inny styl po najechaniu kursorem (:hover) – w nowym stylu nie trzeba dodać transition
 - nadać inny styl za pomocą JavaScript
- **Ogólna postać:**
transition: <property> <duration> <timing-function> <delay>;

Przykład:

```
<head>
  <meta charset="utf-8">
  <style>
    .menu { background: whitesmoke; } /* tło menu */
    /* poniższy zapis zmienia styl elementów div wewnątrz znacznika o class="menu" */
    .menu div { width: 200px; height: 100px; margin: 10px; display: inline-block;
                background: wheat; transition: 2s; }
    .menu div:hover { background: violet; } /* zmiana koloru po najechaniu */
  </style>
</head>
<body>
  <div class="menu">
    <div>A</div>
    <div>B</div>
    <div>C</div>
    <div>D</div>
  </div>
</body>
```

Wynik:



Zadanie 2

W celu realizacji zadania należy:

- Utworzyć tabelę o dwóch kolumnach i pięciu wierszach (zawartość dowolna)
- Dodać do tabeli (**table**) następujące właściwości: `border-collapse: collapse; border: 1px solid silver;`
- Zmienić tło wierszy tak, aby wiersze **nieparzyste** miały inny kolor tła niż wiersze **parzyste**
- Dodać styl zmieniający tło oraz kolor tekstu po najechaniu kursorem na dany wiersz tabeli (`:hover`)
- Dodać efekt przejścia (transition) o wartości czasu 2s (dla znacznika **tr**). Zaobserwować animację występującą po najechaniu kursorem na dowolny wiersz tabeli)
- Dodać atrybut klasy (z dowolną nazwą) dla pierwszego wiersza tabeli (atrybut **class** pierwszego znacznika **tr** bezpośrednio wewnątrz tabeli). Odwołać się w sekcji `<style></style>` do wszystkich znaczników **td** (lub **th**) występujących wewnątrz tego wiersza: `.nazwa td` i zmienić kolor tekstu oraz tła.
- Zastanowić się jak zmienić styl tak, aby efekt najechania kursorem na pierwszy wiersz tabeli nie był widoczny (opcjonalnie).

Ułożenie elementów na stronie

Elementy pływające:

- `float: left; /* ulóż element z lewej strony */`
- `float: right; /* ulóż element z prawej strony */`
- `float: none; /* element nie będący elementem pływającym */`

Wyśrodkowanie elementu wewnątrz (jeżeli jest elementem np. `inline-block`):

- `margin: 0 auto;`

Zmiana położenia elementu:

- `position: static; /* element ułożony domyślnie */`
- `position: relative; /* element przesunięty względem domyślnego położenia – przesunięcie można określić przez np. left: 20px; top: 20px; */`
- `position: absolute; /* element przesunięty względem położenia poprzedniego elementu o position różnym od wartości static */`
- `position: fixed; /* pozycja ustalona na stałe ustalone np. przez wartości: left, top, bottom, right */`
- `position: sticky; /* położenie domyślne, jednak gdy element nie jest widoczny na stronie zachowuje się jak fixed i zmienia położenie na podane przez left, top, bottom, right */`

Zadanie 3

W celu realizacji zadania należy przetestować poniższy kod i zastanowić się nad jego budową.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <style>
      body { margin: 0; padding: 0; background: #777; font-family: 'Trebuchet MS', sans-serif; }
      .header div { max-width: 700px; background: #FFF; margin: 0 auto; height: 50px; user-select: none; }
      .header p { font-size: 26px; padding: 8px; color: #111; margin: 0; padding-left: 30px; }
      .menu { width: 100%; background: #222; }
      .menu div { max-width: 700px; margin: 0 auto; }
      .menu a { display: inline-block; width: 100px; text-align: center; transition: 0.5s;
        text-decoration: none; margin: 5px; color: #FFF; font-weight: 700; }
      .menu a:hover { background: #E32; color: #000; }
      .main div { max-width: 700px; min-height: 500px; margin: 0 auto; background: #FFF; }
      .main div div { padding: 10px; }
    </style>
  </head>
  <body>
    <div class="header">
      <div><p><strong> Nazwa strony</strong></p></div>
    </div>
    <div class="menu">
      <div><a href="#">START</a><a href="#">ARTYKUŁY</a><a href="#">GALERIA</a><a href="#">KONTAKT</a></div>
    </div>
    <div class="main">
      <div>
        <div>Treść</div>
      </div>
    </div>
  </body>
</html>
```