



Nazwa implementacji: CSS i box model

Autor:

Opis implementacji: Poznajemy podstawy CSS oraz dowiadujemy się o rozmieszczaniu elementów na stronie.

Wprowadzenie

CSS (kaskadowe arkusze stylów, ang. Cascading Style Sheets) to sposób opisywania wyglądu treści umieszczonej w dokumencie HTML. Regułki CSS umieszczamy zwykle w oddzielnym pliku z rozszerzeniem .css i dołączamy do dokumentu HTML za pomocą deklaracji:

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

umieszczonej w sekcji <head>.

Regułki

Regułki CSS tworzymy podając tzw. selektor (określający, o jakich elementach mówimy) i atrybuty:

```
selektor {  
    atrybut-pierwszy: wartość;  
    atrybut-drugi: inna wartość;  
}
```

Jak widać, atrybuty są objęte nawiasami klamrowymi i zakończone średnikami.

Selektory

Selektory określają, do jakich elementów odwołują się następujące po nich regułki. Korzystają one w tym celu z nazw tagów, klas (atrybuty class) oraz identyfikatorów (atrybuty id) elementów dokumentu HTML:

```
/* wszystkie linki mają być zielone */  
a {  
    color: green;  
}  
/* elementy o klasie notatka mają mieć żółte tło */  
.notatka {  
    background-color: yellow;  
}  
/* element o id kwadrat ma mieć czerwone tło (wypełnienie)  
i wymiary 100x100px */  
#kwadrat {  
    background-color: red;  
    width: 100px;  
    height: 100px;  
}
```

Parametry CSS: display

Do podstawowych parametrów CSS należy display, który przyjmuje wartości:

block

- element jest umieszczony od nowej linii i rozciąga się na całą szerokość strony,

inline - element pozostaje w tej samej linii i zajmuje tylko tyle miejsca, ile potrzebuje treść w nim zawarta,

• **none - element nie jest widoczny w przeglądarce.**

1

Część tagów ma domyślnie ustawiony parametr

display

na wartość

block

(np. p, div, article) podczas gdy inne używają

inline

(np. span, a, em).

Zadanie:

Zmień nazwę zmodyfikowanego pliku przyklad2.html z poprzedniej lekcji na index.html. Przetestuj działanie parametru display. Żeby to zrobić dodaj jednemu z akapitów klasę

moj-akapit

i utwórz w pliku

style.css

reguły:

```
li {  
  display: inline;  
}  
.moj-akapit {  
  display: none;  
}
```

W pliku index.html dodaj plik style.css jako arkusz stylów dla tego dokumentu (patrz: początek tej lekcji). Zwróć uwagę na to jak zmiana parametru display dla elementów list z domyślnego block na inline zmieniła wygląd list. Na końcu usuń regułki z pliku CSS oraz usuń klasę moj-akapit z akapitu.

Parametry CSS: width i margin

Elementom, które mają ustawioną wartość display: block, możemy nadawać określoną szerokość korzystając z parametru: width: 900px; a następnie umieścić na środku strony korzystając z parametru margin: 0 auto;.

Uwaga: Do uzyskania efektu wyśrodkowania tekstu używamy parametru text-align: center; i należy go odróżniać od zabiegu wyśrodkowania elementu na stronie.

Zadanie:

Nadaj elementowi

<article>

klasę

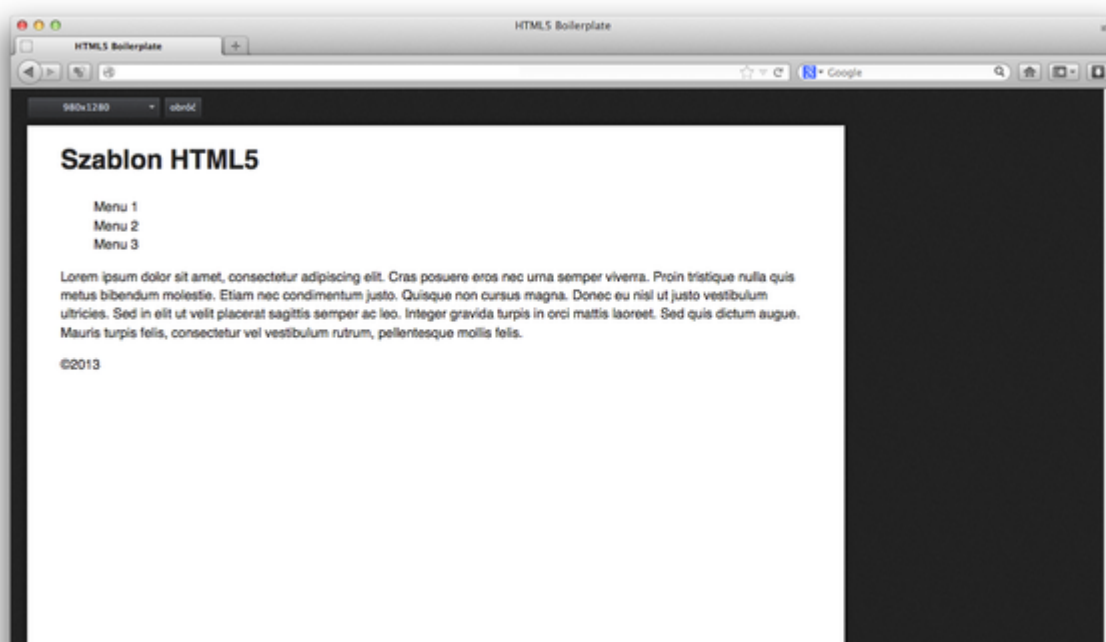
.content

, do dokumenty dodaj plik CSS z regułami:

```
.content {
width: 500px;
margin: 0 auto;
background-color: #eee;
}
```

a następnie

a następnie porównaj zachowanie dokumentu przy zmianie szerokości okna przeglądarki (warto wykorzystać tutaj narzędzie: Widok responsywny) zmieniając parametr width na max-width.



Korzystając z parametru max-width, pozwalamy przeglądarce na zmniejszenie wielkości obiektu, kiedy nie mieści się w oknie, gdy użytkownik korzysta z komputera o mniejszej rozdzielczości.

CSS Box Model

Przed chwilą użyty parametr margin, razem z width, height, oraz dwoma nowymi: padding, border służą do określania parametrów tzw. box modelu:

- margin – określa zewnętrzną odległość od elementu do jego rodzica,
- padding – określa wewnętrzną odległość od elementu do jego zawartości,
- border – określa grubość, styl oraz kolor ramki elementu.

Parametry margin i padding mogą przyjmować od 1 do 4 różnych wartości, które odpowiednio rozkładają się na każdą ze stron. W razie potrzeby można też skorzystać z ich odpowiedników: margin-top, margin-right, margin-bottom, margin-left, oraz: padding-top, padding-right, padding-bottom, padding-left, określających odległość tylko dla jednej ze stron.

3

Zadanie: Do dokumentu HTML dodaj dwa elementy <div> z jednym krótkim paragrafem tekstu i nadając im parametry class="small", class="big" przypisz poniższe właściwości CSS.

```
.small {
width: 500px;
margin: 0 auto;
border: 3px solid blue;
padding: 5px;
}
```

```
.big {
width: 700px;
margin: 0 auto;
border: 20px solid gray;
padding: 20px 60px;
}
```

Następnie zmieniając wartości dla parametrów

width, border i padding

spraw, żeby oba elementy zajmowały taką samą szerokość strony, posiadając różne wartości.

Uwaga: Parametr width nie definiuje ostatecznej szerokości elementu w przeglądarce, a jedynie jej wewnętrzną wartość. Żeby uzyskać rzeczywistą szerokość elementu, należy do wartości width zsumować pozostałe: border i padding.

W trakcie zadania wykorzystaj narzędzia dla twórców witryn dostępne w przeglądarce.



Parametr CSS: float

Kolejnym bardzo często używanym parametrem jest float, który może przyjmować wartości left, right, oraz domyślnie: none. Można użyć go na przykład do stworzenia efektu oblewania zdjęcia tekstem.

Zadanie: W pliku HTML przed paragrafem tekstu dodaj tag wskazujący na dowolne zdjęcie z sieci (wykorzystujemy tylko zdjęcia na licencji CC), a następnie dodając odpowiednie parametry, stwórz efekt oblewania tekstu:

```
.zdjecie {
margin: .5em 1em;
float: left;
}
4
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla sed turpis lectus. Nulla tempus ornare aliquet. Sed ut arcu ut arcu convallis blandit nec non mauris. Vestibulum augue mi, elementum sed condimentum id, dapibus vel libero. Donec tempus mauris vel enim tristique nec molestie eros commodo. Etiam a lacus in eros ultricies aliquam. Fusce condimentum pellentesque mauris, nec bibendum nulla bibendum sit amet. Aliquam malesuada vulputate sem vel bibendum. Donec nec leo turpis, a aliquam dolor. Cras sit amet lacus enim, quis cursus tellus. Aliquam et augue vitae sapien cursus pulvinar ac ac eros. Ut eget tellus at dui elementum imperdiet eget ac lorem. Aliquam feugiat, tellus at mattis viverra, lacus eros viverra arcu, ac cursus eros turpis nec nisl. Vivamus non metus ac nunc accumsan mollis a sit amet mauris. Aenean in convallis mauris.

Następnie wykorzystaj parametry float i margin do rozmieszczenia elementów `<nav>` i `<article class="content">` obok siebie. Ustaw kolory dla nagłówka oraz menu nawigacyjnego strony. Aby nadać tło tylko nagłówkowi strony, a nie wszystkim elementom `<header>` (włącznie z nagłówkami artykułów) użyj selektora `body > header`. Niech menu nawigacyjne ma 200px szerokości. Ustaw elementowi odpowiedni lewy margines dla `.content` tak, aby jego treść nie opływała menu, a była równomiernie oddzielona od lewego brzegu strony.

```
nav {
  float: ?;
  width: ?;
  background-color: #fdd;
}
.content {
  margin-left: ?;
  margin-right: ?;
}
```

Parametr CSS: clear

W przypadku gdy na dużej liczbie elementów ustawimy parametr float, część pozostałych elementów może zacząć na nie nachodzić. W takim przypadku pomocnym parametrem jest clear, który wymusza na elemencie, przejście do nowej linii.

Zadanie:

Dodaj do pliku index.html (pod

`<footer>`)

następujące tagi:

```
<div class="box">Mały blok</div>
<div class="box">Mały blok</div>
<div class="box-wide">Szeroki blok</div>
```

oraz parametry CSS:

5



```
.box {
  border: 3px solid #3a87ad;
  width: 100px;
  height: 100px;
  margin: 0 5px;
  float: left;
}
.wide {
  border: 3px solid #3a87ad
}
```



Następnie uzupełnij dla jednego z elementów parametr `clear: left;` tak żeby elementy rozłożyły się w następujący sposób:



Porównaj zachowanie elementów, zmieniając wartości dla `float` z `left` na `right` i `clear` na `right` lub `both`.

Zadanie:

Usuń elementy i regułki dodane w poprzednim zadaniu. Dodaj kolor tła dla elementu `<footer>`. Spraw by menu nawigacyjne strony było dłuższe od bloku z treścią (elementu

```
.content
```

) i zaobserwuj jak zachowuje się stopka strony. Następnie dodaj regułkę

```
clear: both;
```

dla stopki i zobacz, jak zmienił się układ strony.

Dodatkowa wartość dla CSS `display: inline-block`



Innym sposobem na rozwiązanie poprzedniego problemu jest wykorzystanie display: inline-block;; jednak w przeciwieństwie do float, nie ma możliwości przemieszczenia elementów do prawej strony.

Zadanie: Usuń wszystkie parametry **float** i **clear**, zamiast nich zastosuj **display: inline-block**; tak żeby rozmieszczenie elementów pozostało bez zmian.

Position: absolute, fixed i relative

Ostatnim parametrem, który pozwala na swobodne rozmieszczanie elementów na stronie, jest

position

, który przyjmuje domyślnie wartość

static

, oraz

absolute, relative, fixed.

Ustawienie parametru **position: relative**; niczego samo w sobie nie zmienia w zachowaniu elementu, ale daje możliwość wykorzystania dodatkowych parametrów **top**, **bottom**, **left**, **right**, które pozwalają na przesunięcie elementu w dowolną stronę.

Zadanie: Stwórz dwa elementy <div> i nadaj im poniższe parametry CSS.

```
position: relative;
```

```
position: relative;
top: -30px;
left: 50px;
```

Kolejna wartość position: fixed; pozwala na umieszczenie elementu przy dowolnej krawędzi okna przeglądarki.

Zadanie: Dla jednego z parametrów zmień wartość position na fixed i umieść w lewym górnym rogu używając top: 0px; i left: 0px;.

Parametr position: absolute; działa podobnie co position: fixed; tylko, że zamiast umieszczać element przy krawędzi okna przeglądarki, brane są pod uwagę inne elementy okalające go i posiadające dowolną inną wartość niż position: static;. Najczęściej wykorzystuje się to w połączeniu z position: relative;.

Zadanie: Do dokumentu HTML dodaj następujące tagi:

```
<section class="relative">
  <div class="absolute top left"></div>
  <div class="absolute top right"></div>
  <div class="absolute bottom left"></div>
  <div class="absolute bottom right"></div>
</section>
```

7



a następnie uzupełnij następujący kod CSS tak, żeby każdy z elementów był <div> w oddzielnym rogu.

```
<section>.  
.relative {  
  position: relative;  
  border: 3px solid #3a87ad;  
}  
.absolute {  
  background-color: #999;  
  ???  
}  
.top {  
  ???  
}  
.bottom {  
  ???  
}  
.left {  
  ???  
}  
.right {  
  ???  
}
```

Uwaga: Wysokość w klasie .relative jest ustalona na 4em. Jednostka em jest długością zależną od wielkości użytej czcionki, efektem jej zastosowania będzie skalowanie się wysokości elementu opisanego tą klasą razem ze zmianą wielkości czcionki w przeglądarce (Ctrl++/Ctrl+-/Ctrl+0 dla Mozilla Firefox). Zamiast 4em można podać wielkość w pikselach (np. 100px), wtedy wysokość elementu będzie niezależna od wielkości czcionki.

Sprawdź co się stanie z tagami <div>, gdy <section> będzie miał ustawioną wartość position: static;.

