

# Lekcja 2: Jak działa CSS?

W tej lekcji nauczysz się jak stworzyć swój pierwszy arkusz stylów. Poznasz podstawy modelu CSS i jaki kod jest potrzebny aby CSS działał w dokumencie HTML.

Wiele właściwości używanych w Kaskadowych Arkuszach Stylów (CSS) jest podobnych do tych w HTML. Więc, jeżeli jesteś przyzwyczajony do stosowania HTML jako układ graficzny, najprawdopodobniej rozpoznasz wiele z kodów. Popatrzmy na konkretny przykład.

## Podstawa składni CSS

Powiedzmy że chcemy ładny czerwony kolor tła na stronie:

Używając **HTML** możemy zrobić to tak:

```
<body bgcolor="#FF0000">
```

Dzięki **CSS** ten sam rezultat można osiągnąć w taki sposób:

```
body {background-color: #FF0000;}
```

Jak już pewnie zauważyłeś, kod jest bardzo podobny dla HTML i CSS. Powyższy przykład pokazuje także fundamentalny model CSS:

```
selector {property: value;}
```

Do jakich znaczników HTML stosuje się właściwość (np. "body")

Właściwość oznacza na przykład kolor tła ("background-color")

Wartość właściwości koloru tła może na przykład oznaczać kolor czerwony ("FF0000")

Ale gdzie umieszczam mój kod CSS? To jest dokładnie to czym się teraz zajmujemy.

## Dodawanie CSS do dokumentu HTML

Są trzy sposoby stosowania CSS w dokumentach HTML. Te metody są przedstawione poniżej. Sugerujemy abyś skupił się na trzeciej metodzie tzn. zewnętrznej.

### Metoda 1: W linii (atrybut style)

Pierwszym sposobem jest użycie atrybutu `style`. Poniżej umieszczony jest przykład użycia tej metody do zmiany koloru tła elementu na czerwony:

```

<html>
  <head>
    <title>Example</title>
  </head>
  <body style="background-color: #FF0000;">
    <p>This is a red page</p>
  </body>
</html>

```

## Metoda 2: Wewnętrzna (znacznik style)

Innym sposobem jest umieszczenie kodu CSS dzięki użyciu znacznika HTML - `<style>`. Przykład poniżej:

```

<html>
  <head>
    <title>Example</title>
    <style type="text/css">
      body {background-color: #FF0000;}
      h1 { color: #990000;
          background-color: #FC9804;
        }
    </style>
  </head>
  <body>
    <p>This is a red page</p>
    <h1>Moj pierwszy styl CSS </h1>
  </body>
</html>

```

## Metoda 3: Zewnętrzna (odnośnik do arkusza stylów)

Sugerowaną metodą jest umieszczenie odnośnika do tak zwanego zewnętrznego arkusza stylów. Przykłady w tym kursie będą wykorzystywały właśnie tą metodę.

Zewnętrzny arkusz stylów jest po prostu plikiem tekstowym z rozszerzeniem `.css`. Tak jak każdy inny plik, możesz umieścić arkusz stylów na swoim serwerze internetowym lub na dysku twardym.

Na przykład, powiedzmy że arkusz stylów nazywa się **arkusz.css** i znajduje się w folderze **style**. Sytuację można zilustrować następująco:



Sztuką jest utworzenie odnośnika z dokumentu HTML (default.htm) do arkusza stylów (arkusz.css). Taki odnośnik można stworzyć jedną linią kodu HTML:

```

<link rel="stylesheet" type="text/css" href="style/arkusz.css" />

```

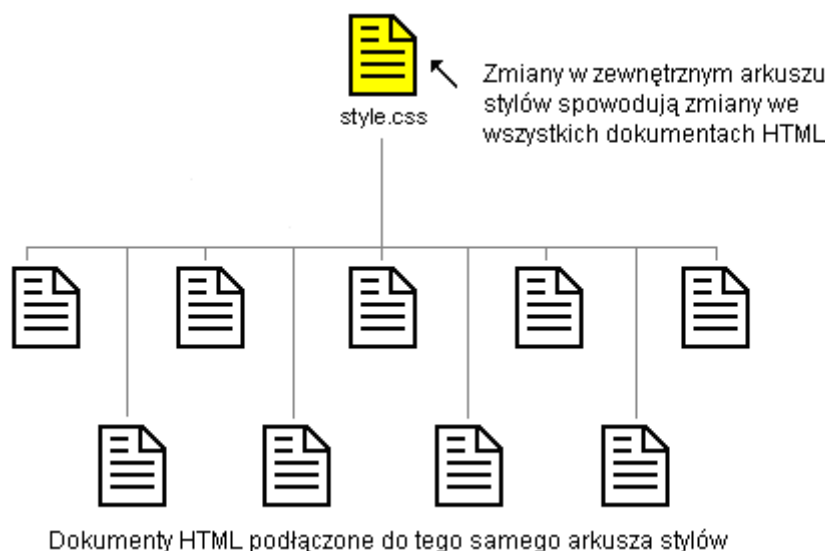
Zauważ jak określiliśmy ścieżkę do arkusza stylów za pomocą atrybutu `href`.

Linia kodu musi zostać umieszczona w sekcji nagłówkowej dokumentu HTML tzn. pomiędzy znacznikiem `<head>` i `</head>`. Tak jak tutaj:

```
<html>
  <head>
    <title>Mój dokument</title>
    <link rel="stylesheet" type="text/css" href="style/arkusz.css" />
  </head>
  <body>
    ...
```

Ten odnośnik mówi przeglądarce, że powinna korzystać z układu graficznego zdefiniowanego w pliku CSS podczas wyświetlania dokumentu HTML.

Naprawdę mądrą rzeczą jest to, że wiele dokumentów HTML może zostać połączonych z tym samym plikiem arkusza stylów. Innymi słowy, jeden plik CSS może kontrolować układ graficzny w wielu dokumentach HTML.



Ta technika zaoszczędzi ci mnóstwo czasu. Jeżeli, na przykład, będziesz chciał zmienić kolor tła strony internetowej zawierającej 100 dokumentów HTML, arkusz stylów uchroni cię od ręcznej zmiany wszystkich 100 stron. Używając CSS, zmianę można zrobić w parę sekund tylko poprzez modyfikacje w centralnym arkuszu stylów.

Poćwiczmy teraz wszystko czego się nauczyliśmy.

## Sprawdź to sam

Otwórz Notatnik (lub jakiegokolwiek edytor tekstowy) i stwórz dwa pliki - plik HTML oraz plik CSS - z następującą zawartością:

## default.htm

```
<html>
  <head>
    <title>Mój dokument</title>
    <link rel="stylesheet" type="text/css" href="style.css" />
  </head>
  <body>
    <h1>Mój pierwszy arkusz stylów</h1>
  </body>
</html>
```

## style.css

```
body {
  background-color: #FF0000;
}
```

Teraz umieść oba pliki w tym samym folderze. Pamiętaj aby zapisać pliki z poprawnymi rozszerzeniami (odpowiednio ".htm" oraz ".css")

Otwórz **default.htm** w swojej przeglądarce i zobacz jak strona uzyskała czerwone tło. Gratulacje! Zrobiłeś swój pierwszy arkusz stylów!

Pospiesz się do [następnej lekcji](#) gdzie popatrzymy na parę właściwości w CSS

# Lekcja 3: Kolory i tła

W tej lekcji nauczysz się stosowania CSS do zmiany kolorów oraz kolorów tła na swojej stronie internetowej. Popatrzymy także na zaawansowane metody pozycjonowania i kontroli obrazków w tle. Opiszemy następujące właściwości CSS:

- [color](#)
- [background-color](#)
- [background-image](#)
- [background-repeat](#)
- [background-attachment](#)
- [background-position](#)
- [background](#)

## Kolor pierwszoplanowy: właściwość 'color'

Właściwość `color` opisuje kolor pierwszoplanowy (najczęściej kolor napisów) elementu.

Na przykład, wyobraź sobie że chcesz aby nagłówki na stronie miały ciemno czerwony kolor. Nagłówki są oznaczone znacznikiem `<h1>` w kodzie HTML. Kod CSS poniżej ustawia kolor elementów `<h1>` na czerwony.

```
h1 {  
    color: #ff0000;  
}
```

- [Pokaż przykład](#)

Kolory mogą zostać wprowadzone jako wartość szesnastkowa tak jak w przykładzie powyżej (#ff0000). Możesz także skorzystać z angielskich nazw najbardziej popularnych kolorów ("red" - czerwony) lub wartości rgb (rgb(255,0,0)).

## Właściwość 'background-color'

Właściwość `background-color` opisuje kolor tła elementów.

Element `<body>` zawiera całą zawartość dokumentu HTML. Więc, aby zmienić kolor tła dla całej strony, właściwość `background-color` powinna stosować się do elementu `<body>`.

Możesz także zastosować kolory tła dla innych elementów, także dla nagłówków i tekstu. W poniższym przykładzie zastosowano różne kolory tła dla elementu `<body>` oraz `<h1>`.

```
body {  
    background-color: #FFCC66;  
}  
  
h1 {  
    color: #990000;  
    background-color: #FC9804;  
}
```

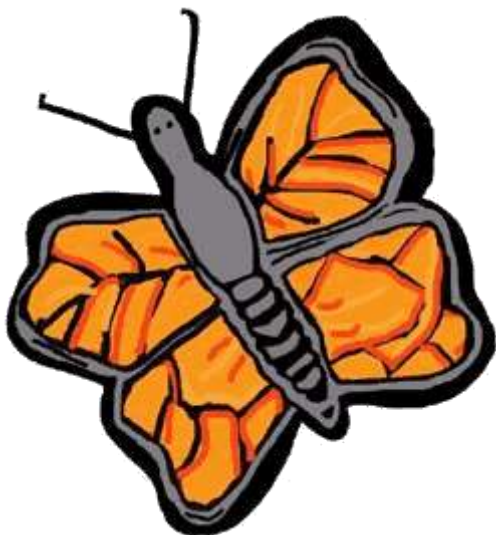
- [Pokaż przykład](#)

Zauważ użycie dwóch właściwości dla `<h1>` rozdzielonych średnikiem.

## Obrazki w tle [background-image]

Właściwość CSS `background-image` jest używana do wstawiania obrazka w tle.

Jako przykład obrazka w tle poniżej użyliśmy motyla. Możesz zapisać obrazek na dysku tak byś mógł sam go użyć w praktyce (kliknij prawym klawiszem i wybierz "zapisz obrazek jako"), lub możesz użyć innego obrazka do ćwiczeń.



Aby umieścić obrazek motyla jako obrazek w tle dla strony internetowej, po prostu zastosuj właściwość `background-image` do elementu `<body>` i określ położenie obrazka.

```
body {  
    background-color: #FFCC66;  
    background-image: url("butterfly.gif");  
}  
  
h1 {  
    color: #990000;  
    background-color: #FC9804;  
}
```

- [Pokaż przykład](#)

Zauważ jak określona jest lokalizacja obrazka jako `url("butterfly.gif")`. To oznacza że obrazek mieści się w tym samym folderze gdzie arkusz stylów. Możesz także odwołać się do obrazka w folderze używając `url("../images/butterfly.gif")` lub nawet w Internecie określając pełen adres do pliku: `url("http://www.html.net/butterfly.gif")`.

## Powtarzanie obrazka w tle [`background-repeat`]

W powyższym przykładzie, czy zauważyłeś że standardowo obrazek powtarzał się w poziomie i w pionie i zakrywał całą stronę? Właściwość `background-repeat` kontroluje to zachowanie.

Tabela poniżej przedstawia cztery możliwe wartości dla właściwości `background-repeat`.

Value	Description	Example
<code>Background-repeat: repeat-x</code>	Obrazek powtarza się w poziomie	<a href="#">Pokaż przykład</a>
<code>background-repeat: repeat-y</code>	Obrazek powtarza się w pionie	<a href="#">Pokaż przykład</a>

<code>background-repeat: repeat</code>	Obrazek powtarza się w poziomie oraz w pionie	<a href="#">Pokaż przykład</a>
<code>background-repeat: no-repeat</code>	Obrazek nie powtarza się	<a href="#">Pokaż przykład</a>

Na przykład, aby uniknąć powtarzania się obrazka w tle kod powinien wyglądać następująco:

```
body {
    background-color: #FFCC66;
    background-image: url("butterfly.gif");
    background-repeat: no-repeat;
}

h1 {
    color: #990000;
    background-color: #FC9804;
}
```

- [Pokaż przykład](#)

## Blokada obrazka w tle [background-attachment]

Właściwość `background-attachment` określa czy obrazek w tle jest stały czy też przewija się wraz z zawierającym go elementem.

Stały obrazek w tle nie poruszy się z tekstem kiedy czytelnik przewija stronę, podczas gdy odblokowany obrazek w tle przewinie się wzdłuż tekstu strony internetowej.

Tabela poniżej przedstawia dwie różne wartości dla `background-attachment`. Kliknij na przykład aby zobaczyć różnice między wartością `scroll` a `fixed`.

Value	Description	Example
<code>Background-attachment: scroll</code>	Obrazek porusza się razem ze stroną - odblokowany	<a href="#">Pokaż przykład</a>
<code>Background-attachment: fixed</code>	Obrazek jest zablokowany	<a href="#">Pokaż przykład</a>

Na przykład, kod poniżej zablokuje obrazek w tle.

```
body {
    background-color: #FFCC66;
    background-image: url("butterfly.gif");
    background-repeat: no-repeat;
    background-attachment: fixed;
}

h1 {
    color: #990000;
    background-color: #FC9804;
}
```

- [Pokaż przykład](#)

## Pozycja obrazka w tle [background-position]

Standardowo, obrazek w tle zostanie umieszczony w górnym lewym rogu ekranu.

Właściwość `background-position` pozwala na zmianę standardowych ustawieni i pozycjonowanie obrazka gdziekolwiek chcesz na ekranie.

Istnieje wiele sposobów ustawienia wartości dla `background-position`. Jednak, wszystkie są formatowane na zbiór koordynatów. Na przykład, wartość `'100px 200px'` ustawi obrazek w tle na pozycji 100px od lewej strony oraz 200px od góry okna przeglądarki.

Koordynaty można zapisać także jako wartość procentową szerokości ekranu, ustalone jednostki (piksele, centymetry, itd.) lub możesz użyć słów: `top`, `bottom`, `center`, `left` oraz `right`. Model poniżej ilustruje system:

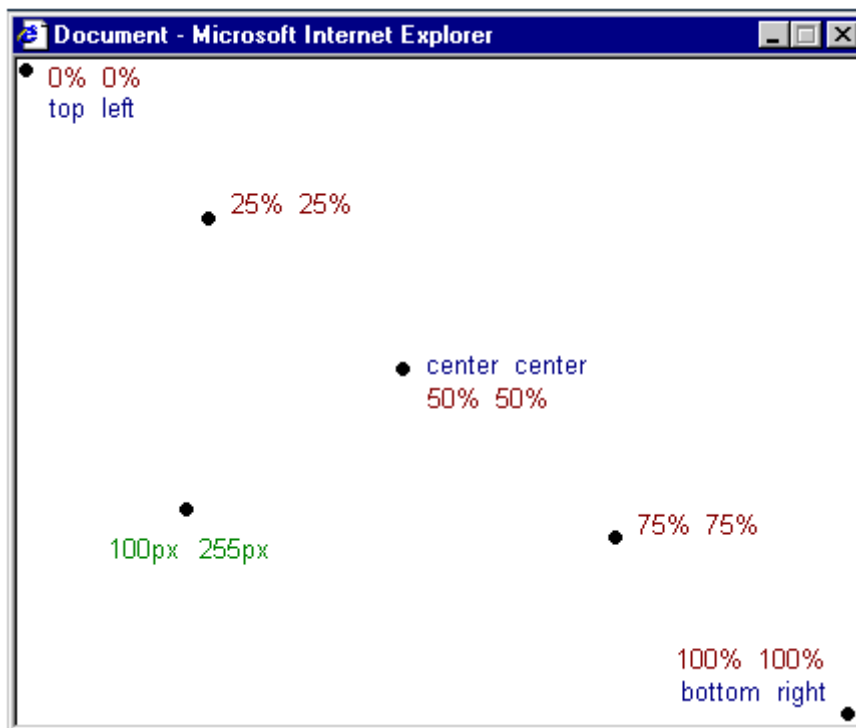


Tabela poniżej przedstawia parę przykładów.

Value	Description	Example
<code>background-position: 2cm 2cm</code>	Obrazek jest ustawiony 2 cm od lewej i 2 cm w dół	<a href="#">Pokaż przykład</a>
<code>background-position: 50% 25%</code>	Obrazek jest ustawiony centralnie w poziomie i w jednej czwartej w pionie licząc od góry	<a href="#">Pokaż przykład</a>
<code>background-position: top right</code>	Obrazek jest ustawiony w górnym prawym rogu strony	<a href="#">Pokaż przykład</a>



Poniższy kod w przykładzie ustawia obrazek w tle w dolnym prawym rogu:

```
body {  
    background-color: #FFCC66;  
    background-image: url("butterfly.gif");  
    background-repeat: no-repeat;  
    background-attachment: fixed;  
    background-position: right bottom;  
}  
  
h1 {  
    color: #990000;  
    background-color: #FC9804;  
}
```

- [Pokaż przykład](#)

## Kompilacja [background]

Właściwość `background` jest skrótem dla wszystkich właściwości tła przedstawionych w lekcji.

Dzięki `background` możesz skompresować wiele właściwości i zapisać swój arkusz stylów w krótszy sposób który ułatwia czytanie kodu.

Na przykład, popatrz na tych parę linii:

```
background-color: #FFCC66;  
background-image: url("butterfly.gif");  
background-repeat: no-repeat;  
background-attachment: fixed;  
background-position: right bottom;
```

Używając `background` ten sam rezultat można zapisać w tylko jednej linijce kodu:

```
background: #FFCC66 url("butterfly.gif") no-repeat fixed right bottom;
```

Lista koment jest następująca :

```
[background-color] | [background-image] | [background-repeat] | [background-attachment] | [background-position]
```

Jeżeli właściwość jest opuszczona, automatycznie zostanie ustawiona na swoją standardową wartość. Na przykład, jeżeli `background-attachment` oraz `background-position` zostaną zabrane z przykładu:

```
background: #FFCC66 url("butterfly.gif") no-repeat;
```

Te dwie właściwości które nie zostały określone teraz ich wartości ustawią się standardowo które jak wiesz to: `scroll` oraz `top left`.

## Podsumowanie

W tej lekcji, nauczyłeś się nowych technik nie możliwych do użycia tylko za pomocą kodu HTML. Zabawę kontynuujemy w [następnej lekcji](#) która przedstawi obszerny zakres możliwości korzystania z CSS do opisywania czcionek.

## Lekcja 4: Czcionki

W tej lekcji nauczysz się o czcionkach i jak nimi manipulować przy pomocy CSS. Omówimy także pewien problem, gdzie wybrana czcionka jest przedstawiana na stronie tylko gdy jest ona zainstalowana w komputerze i przeglądarka ma do niej dostęp. Omówimy następujące właściwości CSS:

- [font-family](#)
- [font-style](#)
- [font-variant](#)
- [font-weight](#)
- [font-size](#)
- [font](#)

### Rodzina czcionki [font-family]

Właściwość `font-family` jest używana do ustawiania listy priorytetowej czcionek które ma wykorzystywać dany element lub cała strona. Jeżeli pierwsza czcionka na liście nie jest zainstalowana na komputerze na którym otwierana jest strona, przeglądarka poszuka kolejnej czcionki na liście dopóki nie znajdzie odpowiedniej.

Są dwa typy nazw używanych do kategoryzowania czcionek: nazwy rodziny i rodzina ogólna. Poniżej wytłumaczenie dokładniejsze wytłumaczenie.

Nazwa rodziny (Family-name)

Przykłady nazwy rodzin czcionek (często zwane "czcionkami") mogą być następujące: "Arial", "Times New Roman" lub "Tahoma".

Rodzina ogólna (Generic family)

Ogólną rodzinę najlepiej przedstawić jako grupę rodzin czcionek z pewnymi określonymi cechami. Przykładem jest sans-serif, który jest kolekcją czcionek bez tzw. "stopy".

Różnice można zilustrować następująco:

Times New Roman  
Garamond  
Georgia

Te trzy rodziny czcionek należą do ogólnej rodziny serif. Charakteryzują się posiadaniem "stopy".

Trebuchet  
Arial  
Verdana

Te trzy rodziny czcionek należą do ogólnej rodziny sans-serif. Charakteryzują się nie posiadaniem "stopy".

Courier  
Courier New  
Andale Mono

Te trzy rodziny czcionek należą do ogólnej rodziny monospace. Charakteryzują się posiadaniem ustalonej szerokości dla wszystkich znaków.

Kiedy robisz już listę czcionek dla swojej strony, naturalnie zaczynaj od najbardziej preferowanej przez siebie czcionki a listę uzupełniaj alternatywnymi czcionkami. Sugerujemy abyś uzupełnił listę rodziną ogólną. Tym sposobem w najgorszym wypadku przeglądarka wykorzysta czcionkę tej samej rodziny jeżeli żadna z wymienionych czcionek nie jest dostępna.

Przykład listy priorytetowej czcionek może wyglądać następująco :

```
h1 {font-family: arial, verdana, sans-serif;}  
h2 {font-family: "Times New Roman", serif;}
```

- [Pokaż przykład](#)

Nagłówki <h1> będą wyświetlone za pomocą czcionki "Arial". Jeżeli czcionka nie jest dostępna na komputerze użytkownika, zostanie użyta czcionka "Verdana". Jeżeli obie czcionki nie będą dostępne, czcionka z rodziny **sans-serif** zostanie użyta do przedstawienia nagłówków.

Zauważ że czcionka "Times New Roman" zawiera spację i z tego powodu zapisana jest w cudzysłowie.

## Styl czcionki [font-style]

Właściwość `font-style` definiuje styl użytej czcionki. Możliwe wartości to: **normal**, **italic** lub **oblique**. W przykładzie poniżej, wszystkie nagłówki oznaczone jako <h2> zostaną zapisane kursywą.

```
h1 {font-family: arial, verdana, sans-serif;}
```

```
h2 {font-family: "Times New Roman", serif; font-style: italic;}
```

- [Pokaż przykład](#)

## Wariant czcionki [font-variant]

Właściwość `font-variant` używamy gdy chcemy wybrać powieźdy czcionką normalną (wartość **normal**) lub kapitalikami (wartość **small-caps**). Czcionka **small-caps** oznacza że będą wykorzystywane mniejszych rozmiarów duże litery zamiast małych liter. Zdziwiony? Popatrz na następujący przykład:

Sans Book SC	Sans Bold SC	Serif Book SC	Serif Bold SC
ABCABC	<b>ABCABC</b>	ABCABC	<b>ABCABC</b>

Jeżeli `font-variant` jest ustawiony na **small-caps** a czcionka o tym wariacie nie jest dostępna dla przeglądarki najprawdopodobniej użyte zostaną zwykłe duże litery.

```
h1 {font-variant: small-caps;}  
h2 {font-variant: normal;}
```

- [Pokaż przykład](#)

## Font weight [font-weight]

Właściwość `font-weight` opisuje jak gruba lub "ciężka" ma być czcionka. Czcionka może być normalna (wartość **normal**) lub pogrubiona (wartość **bold**). Niektóre przeglądarki wspierają nawet użycie liczb między 100-900 (w setkach) do opisu grubości czcionki.

```
p {font-family: arial, verdana, sans-serif;}  
td {font-family: arial, verdana, sans-serif; font-weight: bold;}
```

- [Pokaż przykład](#)

## Wielkość czcionki [font-size]

Wielkość czcionki ustawia się za pomocą właściwości `font-size`.

Istnieje wiele różnych jednostek (np. piksele lub procenty) do wyboru przy opisywaniu wielkości czcionki. W tym kursie skupimy się na najbardziej popularnych jednostkach. Przykład zawiera:

```
h1 {font-size: 30px;}  
h2 {font-size: 12pt;}  
h3 {font-size: 120%;}  
p {font-size: 1em;}
```

- [Pokaż przykład](#)

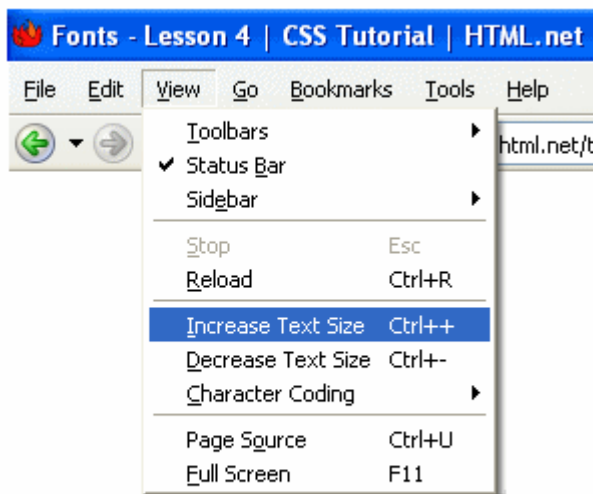
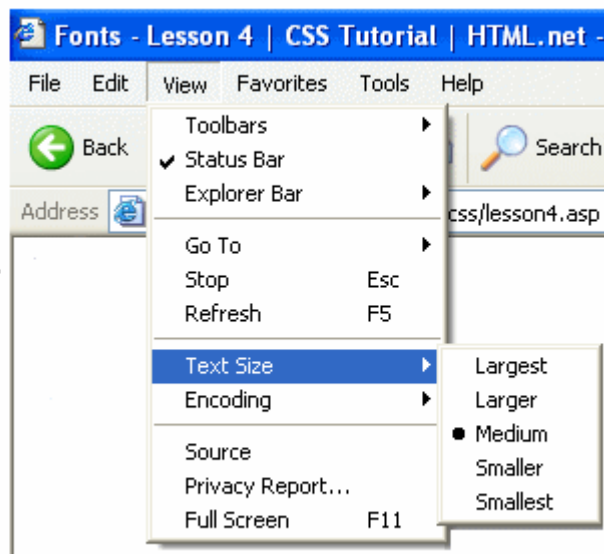
Istnieje jedna kluczowa różnica między czterema powyższymi jednostkami. Jednostka 'px' oraz 'pt' czynią wielkość czcionki absolutną, podczas gdy '%' oraz 'em' pozwalają użytkownikowi określić wielkość czcionki jak im "pasuje". Wiele użytkowników może mieć pewne problemy, być w podeszłym wieku, słabiej widzieć lub po prostu mieć monitor o słabej jakości obrazu. **Aby twoja strona była dostępna dla wszystkich, powinieneś używać jednostek regulowanych takich jak '%' lub 'em'.**

Poniżej przedstawiona jest ilustracja jak regulować wielkość czcionki w przeglądarce Mozilla Firefox oraz Internet Explorer. Spróbuj sam - przydatna funkcjonalność, nie sądzisz?

In most browsers you  
can adjust the text size

Internet Explorer →

Mozilla Firebird



## Kompilacja [font]

Używając właściwości `font` mamy możliwość zapisania wielu różnych właściwości dotyczących czcionki w jedną właściwość.

Na przykład, popatrz na te cztery linie kodu użyte do opisania właściwości czcionki dla akapitu `<p>`:

```
p {  
    font-style: italic;  
    font-weight: bold;  
    font-size: 30px;  
    font-family: arial, sans-serif;  
}
```

Używając właściwości skrótowej, kod będzie uproszczony:

```
p {  
    font: italic bold 30px arial, sans-serif;  
}
```

Kolejność wartości `font` jest następująca:

`font-style` | `font-variant` | `font-weight` | `font-size` | `font-family`

## Podsumowanie

Nauczyłeś się właśnie paru możliwości związanych z czcionkami. Pamiętaj że jedną z podstawowych zalet CSS w określaniu parametrów czcionki jest to, że w każdej chwili możesz zmienić jej właściwości na całej swojej stronie internetowej w zaledwie parę minut. CSS oszczędza twój czas i ułatwia życie. W [następnej lekcji](#) zajmiemy się tekstem.

## Lekcja 5: Tekst

Formatowanie i dodawanie stylu do tekstu jest główną sprawą jaką musi opanować projektant stron internetowych. W tej lekcji zostaniesz wprowadzony w zadziwiające możliwości jakie daje ci CSS gdy chcesz dodać układ graficzny do tekstu. Opiszemy następujące właściwości:

- [text-indent](#)
- [text-align](#)
- [text-decoration](#)
- [letter-spacing](#)
- [text-transform](#)

### Wcięcie tekstu [`text-indent`]

Właściwość `text-indent` pozwala dodać elegancję do akapitów poprzez określenie wcięć dla pierwszej linii akapitu. W poniższym przykładzie zastosowaliśmy wcięcie **30px** dla wszystkich akapitów oznaczonych znacznikiem `<p>`:

```
p {
    text-indent: 30px;
}
```

- [Pokaż przykład](#)

## Ułożenie tekstu [text-align]

Właściwość CSS `text-align` odpowiada za atrybut `align` używany w starej wersji języka HTML. Tekst może układać się albo na lewo (wartość **left**), prawo (wartość **right**) lub też centralnie (wartość **center**). Dodatkowo, wartość **justify** właściwości rozciągnie każdą linię tekstu tak by lewy i prawy margines dla każdej linii był równy. Znasz ten układ na przykładzie gazet i magazynów.

W poniższym przykładzie tekst w nagłówku tabeli `<th>` jest ułożony do prawej strony podczas gdy dane tabeli `<td>` są centrowane. Dodatkowo, normalne akapity tekstowe są justowane:

```
th {
    text-align: right;
}

td {
    text-align: center;
}

p {
    text-align: justify;
}
```

- [Pokaż przykład](#)

## Dekoracja tekstu [text-decoration]

Właściwość `text-decoration` umożliwia dodanie różnej "dekoracji" lub "efektów" do tekstu. Na przykład, możesz podkreślić tekst, umieścić linię przez lub nad tekstem, itd. W kolejnym przykładzie, `<h1>` są podkreślonymi nagłówkami, `<h2>` są nagłówkami z linią nad tekstem a `<h3>` są nagłówkami z linią przechodzącą przez tekst.

```
h1 {
    text-decoration: underline;
}

h2 {
    text-decoration: overline;
}

h3 {
    text-decoration: line-through;
}
```

- [Pokaż przykład](#)

## Odstęp literowy [letter-spacing]

Odstęp między literami tekstu można określić używając właściwości `letter-spacing`. Wartość właściwości jest po prostu porządaną szerokością. Na przykład, jeżeli chcesz umieścić odstępn **3px** między literami w akapicie `<p>` oraz **6px** pomiędzy literami w nagłówku `<h1>` możesz wykorzystać poniższy kod.

```
h1 {
    letter-spacing: 6px;
}

p {
    letter-spacing: 3px;
}
```

- [Pokaż przykład](#)

## Transformacja tekstu [text-transform]

Właściwość `text-transform` kontroluje wielkość liter tekstu. Możesz wybrać wartość: **capitalize**, użyj **uppercase** lub **lowercase** zależnie od tego jak oryginalny tekst jest umieszczony w kodzie HTML.

Przykładem może być słowo "nagłówek" które może zostać przedstawione użytkownikowi jako "NAGŁÓWEK" lub "Nagłówek". Istnieją cztery możliwe wartości właściwości `text-transform`:

### capitalize

Zamienia na dużą literę wszystkie pierwsze litery w wyrazach. Na przykład: "john doe" stanie się "John Doe".

### uppercase

Konwertuje wszystkie litery na duże. Na przykład: "john doe" stanie się "JOHN DOE".

### lowercase

Konwertuje wszystkie litery na małe. Na przykład: "JOHN DOE" stanie się "john doe".

### none

Bez transformacji - tekst jest prezentowany tak jak został on umieszczony w kodzie HTML.

Jako przykład, użyjemy listy nazwisk. Nazwiska oznaczone są znacznikiem `<li>` (list-item). Powiedzmy że chcemy aby nazwiska i imiona zapisane były z pierwszej dużej litery a nagłówki zaprezentować całe tylko dużymi literami.

Popatrz na kod HTML dla tego przykładu i zobaczysz że tekst jest cały umieszczony małą literą.

```
h1 {
    text-transform: uppercase;
}
```



```
}  
  
li {  
    text-transform: capitalize;  
}
```

- [Pokaż przykład](#)

## Podsumowanie

W ostatnich trzech lekcjach nauczyłeś się już wielu właściwości CSS, ale jest jeszcze dużo w CSS do opanowania. W [następnej lekcji](#) popatrzymy bliżej na odnośniki.

## Lekcja 6: Odnośniki

Możesz stosować to czego nauczyłeś się w poprzednich lekcjach dla odnośników (tzn. zmieniać kolory, czcionki, podkreślenia, itd). Nowością jest to że CSS pozwala definiować właściwości zależne od stanu odnośnika, czyli tego czy odnośnik jest aktywny, odwiedzony, nie odwiedzony lub kursor jest nad odnośnikiem. To daje możliwość dodania użytecznych efektów na twojej stronie internetowej. Aby kontrolować te efekty wykorzystuje się tak zwane pseudo-klasy.

### Co to jest pseudo-klasa?

Pseudo-klasa pozwala ci zaangażować w sprawę warunków lub zdarzeń gdy definiujesz właściwości CSS dla znaczników HTML.

Popatrzmy na przykład . Jak już wiesz, odnośniki określamy znacznikiem `<a>`. Możemy więc użyć `a` jako selektor w CSS:

```
a {  
    color: blue;  
}
```

Odnośnik może mieć różne stany. Na przykład, może być odwiedzony lub nie odwiedzony. Możesz użyć pseudo-klas do skojarzenia różnych stylów zależnie od stanów.

```
a:link {  
    color: blue;  
}  
  
a:visited {  
    color: red;  
}
```

Użyj `a:link` oraz `a:visited` odpowiednio dla odnośników nie odwiedzonych oraz odwiedzonych. Odnośniki aktywne mają pseudo-klasę `a:active` oraz `a:hover` jeżeli kursor jest nad odnośnikiem.

Przebrniemy przez każdą z czterech pseudo-klas z przykładami i dokładnym wyjaśnieniem.

## Pseudo-klasa: link

Pseudo-klasa `:link` jest używana dla odnośników prowadzących do stron nie odwiedzonych przez użytkownika.

W przykładzie poniżej, nie odwiedzony odnośnik będzie jasno niebieski.

```
a:link {  
    color: #6699CC;  
}
```

- [Pokaż przykład](#)

## Pseudo-klasa: visited

Pseudo-klasa `:visited` jest używana dla odnośników prowadzących do stron odwiedzonych już przez użytkownika. Na przykład, kod poniżej uczyni wszystkie odwiedzone odnośniki ciemno purpurowe:

```
a:visited {  
    color: #660099;  
}
```

- [Pokaż przykład](#)

## Pseudo-klasa: active

Pseudo-klasa `:active` jest używana dla odnośników aktywnych.

Przykład zamieni kolor tła aktywnych odnośników na żółty:

```
a:active {  
    background-color: #FFFF00;  
}
```

- [Pokaż przykład](#)

## Pseudo-klasa: hover

Pseudo-klasa `:hover` jest używana kiedy kursor myszy najedzie na odnośnik.

Można to wykorzystać do tworzenia interesujących efektów. Na przykład, jeżeli chcemy aby nasze odnośniki były pomarańczowe i pochylone kiedy kursor najedzie na odnośnik, kod CSS powinien wyglądać następująco:

```
a:hover {
```

```
color: orange;
font-style: italic;
}
```

- [Pokaż przykład](#)

## Przykład 1: Efekt kiedy kursor jest nad odnośnikiem

Bardzo popularne jest tworzenie różnych efektów kiedy kursor jest nad odnośnikiem. Popatrzmy więc na kilka dodatkowych przykładów związanych z pseudo-klasą `:hover`.

### Przykład 1a: Odstęp między literami

Jak pewnie pamiętasz z [lekcji 5](#), odstęp między literami można regulować poprzez właściwość `letter-spacing`. Można zastosować tę właściwość do odnośnika jako specjalny efekt:

```
a:hover {
  letter-spacing: 10px;
  font-weight: bold;
  color: red;
}
```

- [Pokaż przykład](#)

### Przykład 1b: UPPERCASE and lowercase

W [lekcji 5](#) patrzyliśmy na właściwość `text-transform`, która zmieniała wielkość liter. To także może zostać użyte jako efekt w odnośniku:

```
a:hover {
  text-transform: uppercase;
  font-weight: bold;
  color: blue;
  background-color: yellow;
}
```

- [Pokaż przykład](#)

Te dwa przykłady ilustrują ideę praktycznie nieskończonych możliwości kombinacji różnych właściwości. Możesz stworzyć swój własny efekt - spróbuj sam!

## Przykład 2: Usuń podkreślenie odnośnika

Często zadawanym pytaniem jest jak usunąć podkreślenie odnośnika ?

**Powinieneś rozpatrzyć uważnie czy jest potrzebne usunięcie podkreślenia, co może znacząco zmniejszyć użyteczność twojej strony.** Ludzie są przyzwyczajeni do niebieskiego podkreślenia odnośników na stronach internetowych i wiedzą że mogą na nie kliknąć. Nawet moja mama wie o tym! Jeżeli zmienisz podkreślenie i kolor odnośnika istnieje duża szansa że

użytkownik zmiesza się i nie zdobędzie pełnych korzyści z treści umieszczonej na twojej stronie.

To smutne, bardzo łatwo jest usunąć podkreślenie z odnośnika. Jak już wiesz z [lekcji 5](#), właściwość `text-decoration` może zostać użyta do określenia czy tekst ma być podkreślony czy też nie. Aby usunąć podkreślenie, po prostu ustaw wartość właściwości `text-decoration` na `none`.

```
a {  
    text-decoration: none;  
}
```

Alternatywnie, możesz ustawić `text-decoration` wraz z innymi właściwościami dla wszystkich czterech pseudo-klas.

```
a:link {  
    color: blue;  
    text-decoration: none;  
}  
  
a:visited {  
    color: purple;  
    text-decoration: none;  
}  
  
a:active {  
    background-color: yellow;  
    text-decoration: none;  
}  
  
a:hover {  
    color: red;  
    text-decoration: none;  
}
```

- [Pokaż przykład](#)

## Podsumowanie

W tej lekcji nauczyłeś się o pseudo-klasach oraz używania w ich obrebie paru właściwości z poprzednich lekcji. To powinno pokazać ci ię możliwości jakie dostarcza CSS.

W [następnej lekcji](#) nauczymy cię jak definiować właściwości dla konkretnego elementu oraz grupy elementów.

## Lekcja 7: Identyfikacja i grupowanie elementów (class i id)

Czasami chcemy zastosować ten sam określony styl do pewnego elementu lub grupy elementów. W tej lekcji, pokażemy bliżej jak można korzystać z atrybutów `class` oraz `id` do określenia właściwości stylu pewnych elementów.

Jak możesz pokolorować dany nagłówek inaczej niż wszystkie inne nagłówki na stronie? Jak możesz pogrupować swoje odnośniki w różne kategorie i dać każdej z kategorii inny styl? Oto przykłady odpowiadające na proste pytania zadane w tej lekcji.

## Grupowanie elementów poprzez klasę

Powiedzmy że mamy dwie listy odnośników do różnych stron związanych z czerwonym i białym winem. Kod HTML może wyglądać następująco:

```
<p>Grona białego wina:</p>
<ul>
<li><a href="ri.htm">Riesling</a></li>
<li><a href="ch.htm">Chardonnay</a></li>
<li><a href="pb.htm">Pinot Blanc</a></li>
</ul>

<p>Grona czerwonego wina:</p>
<ul>
<li><a href="cs.htm">Cabernet Sauvignon</a></li>
<li><a href="me.htm">Merlot</a></li>
<li><a href="pn.htm">Pinot Noir</a></li>
</ul>
```

- [Pokaż przykład](#)

Teraz chcemy żeby odnośniki z kategorii białego wina były żółte, z czerwonego natomiast żeby były czerwone a reszta istniejących odnośników żeby była niebieska.

Aby to osiągnąć, dzielimy odnośniki w dwie kategorie. Robimy tak poprzez nadanie klasy używając atrybutu `class`.

Spróbujmy określić pewne klasy używając kodu z powyższego przykładu:

```
<p>Grona białego wina:</p>
<ul>
<li><a href="ri.htm" class="whitewine">Riesling</a></li>
<li><a href="ch.htm" class="whitewine">Chardonnay</a></li>
<li><a href="pb.htm" class="whitewine">Pinot Blanc</a></li>
</ul>

<p>Grona czerwonego wina:</p>
<ul>
<li><a href="cs.htm" class="redwine">Cabernet Sauvignon</a></li>
<li><a href="me.htm" class="redwine">Merlot</a></li>
<li><a href="pn.htm" class="redwine">Pinot Noir</a></li>
</ul>
```

Możemy teraz zdefiniować specjalne właściwości dla odnośników należących odpowiednio do białego oraz czerwonego wina.

```

a {
    color: blue;
}

a.whitewine {
    color: #FFBB00;
}

a.redwine {
    color: #800000;
}

```

- [Pokaż przykład](#)

Jak pokazaliśmy w przykładzie, możesz zdefiniować właściwości dla elementów które należą do konkretnej klasy poprzez użycie **.nazwaklas** w arkuszu stylów.

## Identyfikacja elementu poprzez id

Dodatkowo aby pogrupować elementy, możesz potrzebować umieć identyfikować jeden unikalny element. Robi się to poprzez użycie atrybutu `id`.

Co jest specjalnego w atrybucie `id`, to że nie mogą istnieć dwa elementy w danym dokumencie HTML o tej samej wartości atrybutu `id`. Każde `id` musi być unikalne. Innymi słowy, powinieneś używać atrybutu `class` jeżeli chcesz grupować parę lementów. Teraz, popatrzmy bliżej na przykład możliwości użycia atrybutu `id`:

```

<h1>Sekcja 1</h1>
...
<h2>Sekcja 1.1</h2>
...
<h2>Sekcja 1.2</h2>
...
<h1>Sekcja 2</h1>
...
<h2>Sekcja 2.1</h2>
...
<h3>Sekcja 2.1.2</h3>
...

```

Powyżej mogą być nagłówki do sekcji akapitów. Naturalnie możemy dodać `id` do każdej z sekcji:

```

<h1 id="c1">Sekcja 1</h1>
...
<h2 id="c1-1">Sekcja 1.1</h2>
...
<h2 id="c1-2">Sekcja 1.2</h2>
...
<h1 id="c2">Sekcja 2</h1>
...
<h2 id="c2-1">Sekcja 2.1</h2>
...

```

```
<h3 id="c2-1-2">Sekcja 2.1.2</h3>
...
```

Powiedzmy że nagłówek dla sekcji 1.2 musi być czerwony. Możesz tak zrobić za pomocą następującego kodu CSS:

```
#c1-2 {
  color: red;
}
```

- [Pokaż przykład](#)

Jak widać w powyższym przykładzie możesz zdefiniować właściwości dla konkretnego elementu poprzez użycie selektora `#id` w arkuszu stylów dokumentu.

## Podsumowanie

W tej lekcji nauczyliśmy się dodatkowej możliwości użycia selektorów, `class` oraz `id`, możesz teraz określić właściwości dla konkretnych elementów.

W [następnej lekcji](#), popatrzymy bliżej na dwa elementy HTML, które często łączymy z właściwościami CSS: `<span>` oraz `<div>`.

# Lekcja 8: Grupowanie elementów (span i div)

Elementy `<span>` i `<div>` są używane do grupowania i strukturyzowania dokumentu. Będziesz je często wykorzystywał wraz z atrybutami `class` oraz `id`.

W tej lekcji, popatrzymy bliżej na użycie `<span>` oraz `<div>` gdyż właśnie te dwa elementy HTML są ściśle związane z CSS.

- Grupowanie za pomocą `<span>`
- Grupowanie za pomocą `<div>`

## Grupowanie za pomocą `<span>`

Element `<span>` możesz nazwać elementem naturalnym gdyż sam w sobie nie dodaje nic do dokumentu. Ale razem z CSS, `<span>` może dodać wiele interesujących cech wizualnych do określonej części tekstu w dokumencie.

Przykładem dla nas niech będzie tekst Benjamina Franklina:

```
<p>Early to bed and early to rise
makes a man healthy, wealthy and wise.</p>
```

Powiedzmy że chcemy aby część tekstu Pana Franklina została pokolorowana wyróżniona na czerwono. Do tego celu, zaznaczamy dany tekst znacznikiem `<span>`. Każdy `span` zawiera w sobie ustawiony atrybut `class`, który potem wykorzystujemy w naszym arkuszu stylów:

```
<p>Early to bed and early to rise  
makes a man <span class="benefit">healthy</span>,  
<span class="benefit">wealthy</span>  
and <span class="benefit">wise</span>.</p>
```

CSS należący do dokumentu:

```
span.benefit {  
    color:red;  
}
```

- [Pokaż przykład](#)

Oczywiście możemy także wykorzystać atrybut `id` aby dodać styl do elementu `<span>`. W tym przypadku pamiętaj tylko że nie wolno umieszczać dwóch takich samych wartości dla atrybutu `id`.

## Grupowanie za pomocą `<div>`

Podczas gdy element `<span>` używamy w obrębie elementów blokowych jak widać było w poprzednim przykładzie, `<div>` używamy do grupowania jednego lub większej ilości elementów blokowych.

Oprócz tej różnicy, grupowanie za pomocą elementu `<div>` działa mniej więcej w ten sam sposób. Popatrzmy na przykład z dwiema listami prezydentów Stanów Zjednoczonych podzielonych dwiema ugrupowaniami politycznymi:

```
<div id="democrats">  
<ul>  
<li>Franklin D. Roosevelt</li>  
<li>Harry S. Truman</li>  
<li>John F. Kennedy</li>  
<li>Lyndon B. Johnson</li>  
<li>Jimmy Carter</li>  
<li>Bill Clinton</li>  
</ul>  
</div>
```

```
<div id="republicans">  
<ul>  
<li>Dwight D. Eisenhower</li>  
<li>Richard Nixon</li>  
<li>Gerald Ford</li>  
<li>Ronald Reagan</li>  
<li>George Bush</li>  
<li>George W. Bush</li>  
</ul>  
</div>
```



Natomiast w arkuszu stylów, określamy grupy w ten sam sposób jak wyżej:

```
#democrats {  
    background:blue;  
}  
  
#republicans {  
    background:red;  
}
```

- [Pokaż przykład](#)

W przykładach powyżej, użyliśmy tylko elementów `<div>` oraz `<span>` wykorzystując tylko właściwości koloru tekstu i tła. Oba elementy mają o wiele większy potencjał. Jednak nie przedstawimy tego w tej lekcji. Popatrzymy na to później w naszym kursie.

## Podsumowanie

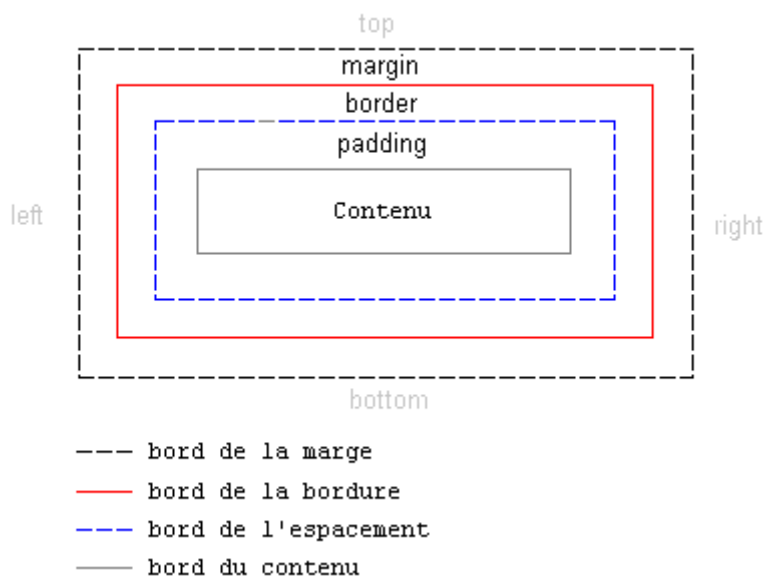
W [lekcji 7](#) oraz lekcji 8, nauczyłeś się o selektorach `id` oraz `class` i o elementach `span` oraz `div`.

Powinieneś teraz potrafić grupować i identyfikować wszystkie części swojego dokumentu, co jest wielkim krokiem w kierunku opanowania CSS. W [lekcji 9](#) wprowadzimy cię w model blokowy.

## Lekcja 9: Model blokowy

Model blokowy w CSS opisuje bloki, które są generowane dla elementów HTML. Model blokowy zawiera także dokładne opcje związane z ustawieniem marginesu zewnętrznego, wewnętrznego, obramowania i zawartością każdego elementu. Poniższy diagram pokazuje jak model blokowy jest skonstruowany:

### Model blokowy w CSS



Powyższa ilustracja może wydawać się bardzo teoretyczna, więc spróbujmy użyć modelu blokowego w konkretnym przypadku z nagłówkami i tekstem. HTML dla naszego przykładu (uniwersalna deklaracja praw człowieka):

```
<h1>Artykuł 1:</h1>
```

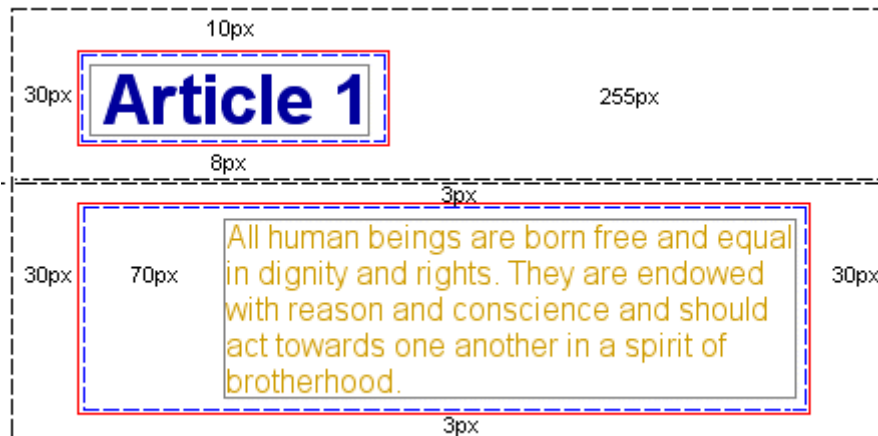
```
<p>Wszystkie istoty ludzkie rodzą się wolne  
i równe w godności i prawach.</p>
```

Poprzez dodanie koloru i informacji i czcionce przykład będzie wyglądał następująco:

# Article 1

All human beings are born free and equal  
in dignity and rights. They are endowed  
with reason and conscience and should  
act towards one another in a spirit of  
brotherhood.

Przykład zawiera dwa elementy : `<h1>` i `<p>`. Model blokowy dla dwóch elementów może być zilustrowany następująco:



Nawet jeżeli wygląda to trochę skomplikowanie, ilustracja pokazuje jak każdy element HTML jest otoczony przez bloki. Bloki które możemy ustawić poprzez CSS.

<% '

Właściwości regulujące bloki to: `padding`, `margin` i `border`. Następne dwie lekcje będą dotyczyć właśnie tych trzech właściwości:

,

- [Lekcja 10: Model blokowy - `margin` & `padding`](#)

,

- [Lekcja 11: Model blokowy - `border`](#)

,

Kiedy ukończysz te dwie lekcje, zapanujesz nad modelem blokowym i będziesz potrafił łatwiej i zgrabniej tworzyć układ graficzny dla swoich dokumentów niż robiłeś to wcześniej w starym stylu za pomocą tabel.

%>

## Podsumowanie

W tej lekcji zostałeś wprowadzony w model blokowy. W kolejnych trzech lekcjach popatrzymy bliżej na to jak tworzyć i kontrolować elementy w modelu blokowym.

## Lekcja 10: Margines wewnętrzny i zewnętrzny

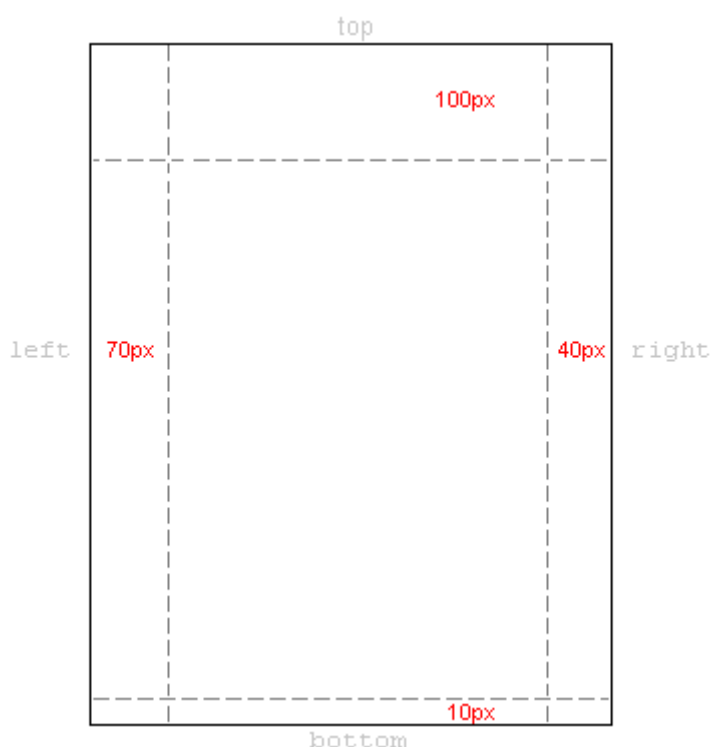
W poprzedniej lekcji zostałeś wprowadzony w model blokowy. W tej lekcji, popatrzymy jak możesz zmienić prezentację elementów poprzez ustawienie właściwości `margin` i `padding`.

- [Ustaw margines zewnętrzny w elemencie](#)
- [Ustaw margines wewnętrzny w elemencie](#)

## Ustaw margines zewnętrzny w elemencie

Element ma cztery strony: prawą, lewą, dolną i górną. `margin` określa odległość każdej ze stron do sąsiedniego elementu (lub obramowania dokumentu). Patrz także na diagram w [lekcji 9](#) dla ilustracji.

Jako pierwszy przykład, popatrzymy jak definiuje się margines zewnętrzny dla dokumentu tzn. dla elementu `<body>`. Poniższa ilustracja przedstawia margines zewnętrzny na stronie jaki sobie wybraliśmy.



Kod CSS dla przykładu będzie wyglądał następująco:

```
body {  
    margin-top: 100px;  
    margin-right: 40px;  
    margin-bottom: 10px;  
    margin-left: 70px;  
}
```

Możesz także wybrać bardziej elegancką kompilację:

```
body {
```

```
margin: 100px 40px 10px 70px;
}
```

- [Pokaż przykład](#)

Możesz ustawić marginesy zewnętrzne w ten sam sposób dla prawie każdego elementu. Na przykład, możesz zdefiniować marginesy zewnętrzne dla wszystkich akapitów zaznaczonych elementem `<p>`:

```
body {
    margin: 100px 40px 10px 70px;
}

p {
    margin: 5px 50px 5px 50px;
}
```

- [Pokaż przykład](#)

## Ustaw margines wewnętrzny w elemencie

Margines wewnętrzny można także zrozumieć jako "wypełnianie". To ma sens gdyż margines wewnętrzny nie dotyczy odległości między elementami a jedynie definiuje wewnętrzną odległość między obramowaniem a zawartością elementu.

Użyteczność marginesu wewnętrznego można zilustrować patrząc na prosty przykład, w którym wszystkie nagłówki posiadają kolor tła:

```
h1 {
    background: yellow;
}

h2 {
    background: orange;
}
```

- [Pokaż przykład](#)

Poprzez zdefiniowanie marginesu wewnętrznego dla wszystkich nagłówków, zmieniasz wielkość wypełniania wokół tekstu dla każdego nagłówka:

```
h1 {
    background: yellow;
    padding: 20px 20px 20px 80px;
}

h2 {
    background: orange;
    padding-left: 120px;
}
```

- [Pokaż przykład](#)

## Podsumowanie

Jesteś teraz na drodze do opanowania modelu blokowego w CSS. W następnej lekcji, popatrzymy bliżej na możliwości ustawiania obramowań w różnych kolorach i kształtach dla elementów.

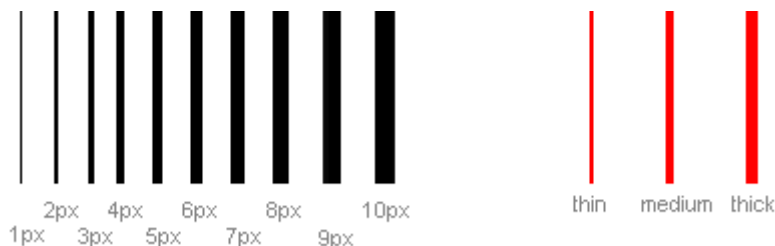
## Lekcja 11: Obramowania

Obramowania można używać w wielu przypadkach, na przykład dekorując element lub podkreślając oddzielenie dwóch rzeczy. CSS daje ci nieskończoną możliwość wykorzystania obramowań na twojej stronie.

- [border-width](#)
- [border-color](#)
- [border-style](#)
- [Przykłady definicji obramowań](#)
- [border](#)

### Szerokość obramowań [border-width]

Szerokość obramowań definiujemy używając właściwości `border-width`, która może otrzymać wartości `thin`, `medium` oraz `thick`, lub wartość numeryczną, określoną w pikselach. Poniższy rysunek przedstawia system:



### Kolor obramowań [border-color]

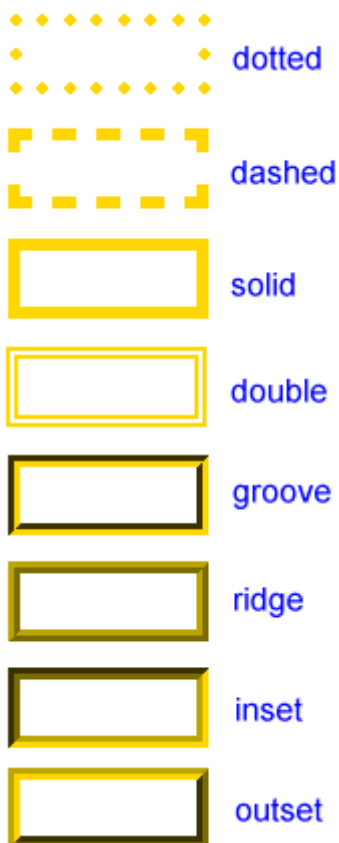


Właściwość `border-color` definiuje jakiego koloru ma być obramowanie. Wartości dopuszczalne we właściwości to np: `"#123456"`, `"rgb(123,123,123)"` lub `"yellow"`.

### Typy obramowań [border-style]

Istnieje wiele typów obramowań do wyboru. Poniżej przedstawiam 8 różnych typów obramowań i ich interpretacje w Internet Explorer 5.5. Wszystkie przykłady wykorzystują kolor złoty `"gold"` oraz grubość `"thick"` ale oczywiście można korzystać z innego koloru lub grubości obramowania.

Jeżeli nie chcesz wyświetlić żadnego obramowania użyj `none` lub `hidden`.



## Przykłady definicji obramowań

Te trzy wyżej opisane właściwości można połączyć przez jeden element i produkować najróżniejsze obramowania. Aby zilustrować to przykładem, popatrzmy na dokument gdzie zdefiniowane są różne obramowania dla elementów `<h1>`, `<h2>`, `<ul>` and `<p>`. Rezultat może nie być zaskakujący ale przedstawia jak wiele możliwości jest do wykorzystania:

```
h1 {  
    border-width: thick;  
    border-style: dotted;  
    border-color: gold;  
}  
  
h2 {  
    border-width: 20px;  
    border-style: outset;  
    border-color: red;  
}  
  
p {  
    border-width: 1px;  
    border-style: dashed;  
    border-color: blue;  
}  
  
ul {  
    border-width: thin;  
    border-style: solid;
```

```
border-color: orange;
}
```

- [Pokaż przykład](#)

Możliwe jest ustawienie konkretnych właściwości oddzielnie dla górnej, dolnej, prawej oraz lewej strony obramowania. Kolejny przykład pokazuje jak skorzystać z tej możliwości:

```
h1 {
  border-top-width: thick;
  border-top-style: solid;
  border-top-color: red;

  border-bottom-width: thick;
  border-bottom-style: solid;
  border-bottom-color: blue;

  border-right-width: thick;
  border-right-style: solid;
  border-right-color: green;

  border-left-width: thick;
  border-left-style: solid;
  border-left-color: orange;
}
```

- [Pokaż przykład](#)

## Kompilacje [border]

Tak jak wiele innych właściwości, także ta może złączyć kilka cech związanych z obramowaniami w jedną właściwość. Popatrzmy na przykład:

```
p {
  border-width: 1px;
  border-style: solid;
  border-color: blue;
}
```

Może być skompilowany w :

```
p {
  border: 1px solid blue;
}
```

## Podsumowanie

W tej lekcji nauczyłeś się o nieskończonych możliwościach jakie daje ci CSS w dziedzinie używania obramowań a stronie.



Następna lekcja wprowadzi cię w definiowanie rozmiarów w modelu blokowym - wysokości i szerokości.

## Lekcja 12: Wysokość i szerokość

Do teraz, nie interesowaliśmy się zbytnio rozmiarami elementów z którymi pracowaliśmy. W tej lekcji, popatrzymy na to jak definiować wysokość oraz szerokość elementów.

- [szerokość](#)
- [wysokość](#)

### Ustawianie szerokości [width]

Dzięki właściwości `width`, możesz zdefiniować szerokość elementu.

Ten prosty przykład przedstawia blok w którym został umieszczony tekst:

```
div.box {  
    width: 200px;  
    border: 1px solid black;  
    background: orange;  
}
```

- [Pokaż przykład](#)

### Ustawianie wysokości [height]

Zauważ jak w przykładzie powyżej wysokość bloku uzależniona jest od wysokości zawartości. Możesz wpłynąć na wysokość elementu dzięki właściwości `height`. Jako przykład spróbujmy ustawić wysokość bloku na 500px:

```
div.box {  
    height: 500px;  
    width: 200px;  
    border: 1px solid black;  
    background: orange;  
}
```

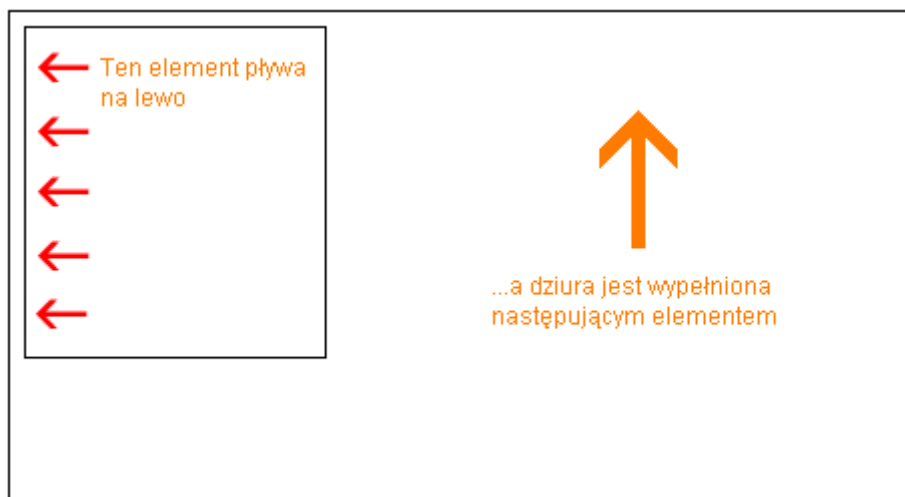
- [Pokaż przykład](#)

## Podsumowanie

[Lekcja 9](#), [10](#), [11](#) oraz [12](#) wprowadziła się w model blokowy CSS. Możesz teraz zobaczyć, jak wiele możliwości daje model blokowy. Możliwe że do teraz używałeś tabel HTML do tworzenia układu graficznego na stronie, ale dzięki CSS i modelowi blokowemu masz już możliwość osiągnięcia eleganckiego i bardziej precyzyjnego układu graficznego zgodnego ze standardem W3C.

# Lekcja 13: Elementy pływające

Elementy mogą ustawić się na lewo lub prawo względem innych przy wykorzystaniu właściwości `float`. Mówi się że blok wraz z zawartością pływa na lewej stronie lub na prawej w dokumencie (lub odpowiednim bloku) (patrz do [lekcji 9](#) dla dokładnego opisu modelu blokowego). Następujący rysunek ilustruje zasadę:



Jeżeli, dla przykładu, chcemy mieć obrazek osłonięty tekstem, rezultat wyglądał by tak:



## Jak to jest zrobione?

Kod HTML dla powyższego przykładu:

```
<div id="picture">
  
</div>

<p>causas naturales et antecedentes,
idcirco etiam nostrarum voluntatum...</p>
```

Aby obrazek ustawiony został na lewej stronie a tekst otaczał go z prawej, musisz jedynie zdefiniować szerokość elementu zawierającego w sobie obrazek oraz ustawić właściwość `float` na wartość `left`:

```
#picture {
    float:left;
    width: 100px;
}
```

- [Pokaż przykład](#)

## Kolejny przykład: kolumny

Pływanie może być także używane do tworzenia kolumn w dokumencie. Aby stworzyć kolumnę, po prostu strukturyzujesz odpowiednie kolumny w kodzie HTML za pomocą znacznika `<div>` jak niżej:

```
<div id="column1">
    <p>Haec disserens qua de re agatur
    et in quo causa consistat non videt...</p>
</div>

<div id="column2">
    <p>causas naturales et antecedentes,
    idcirco etiam nostrarum voluntatum...</p>
</div>

<div id="column3">
    <p>nam nihil esset in nostra
    potestate si res ita se haberet...</p>
</div>
```

Teraz porządaną szerokość kolumny ustawiamy na np. 33%, a potem ustawiamy pływanie każdej kolumny na lewo poprzez właściwość `float`:

```
#column1 {
    float:left;
    width: 33%;
}

#column2 {
    float:left;
    width: 33%;
}

#column3 {
    float:left;
    width: 33%;
}
```

- [Pokaż przykład](#)

`float` może otrzymać wartości: **left**, **right** lub **none**.

## Właściwość clear

Właściwość `clear` używamy do kontroli zachowania kolejnego elementu w dokumencie za elementem pływającym.

Standardowo, kolejne elementy przesuwają się w taki sposób, aby wypełnić puste miejsce stworzone przez element pływający po stronach. Popatrz na przykład powyżej gdzie tekst automatycznie przesuwa się na bok zdjęcia Billa Gatesa.

Właściwość `clear` może przyjmować wartości **left**, **right**, **both** or **none**. Zasada jest następująca, Jeżeli `clear`, dla przykładu, ma wartość `both` dla elementu, górny margines obramowania elementu będzie zawsze poniżej dolnego marginesu obramowania elementu nad nim pływającego.

```
<div id="picture">
  
</div>

<h1>Bill Gates</h1>

<p class="floatstop">causas naturales et antecedentes,
idciro etiam nostrarum voluntatum...</p>
```

Aby wyłączyć pływanie tekstu w stronę obrazka, możemy wykorzystać następujący kod CSS:

```
#picture {
  float:left;
  width: 100px;
}

.floatstop {
  clear:both;
}
```

- [Pokaż przykład](#)

## Podsumowanie

Pływanie jest bardzo użyteczne w wielu sytuacjach i często wykorzystuje się je wraz z pozycjonowaniem. W [następnej lekcji](#) popatrzmy bliżej na pozycjonowanie bloków w sposób relatywny lub absolutny.

## Lekcja 14: Pozycjonowanie elementów

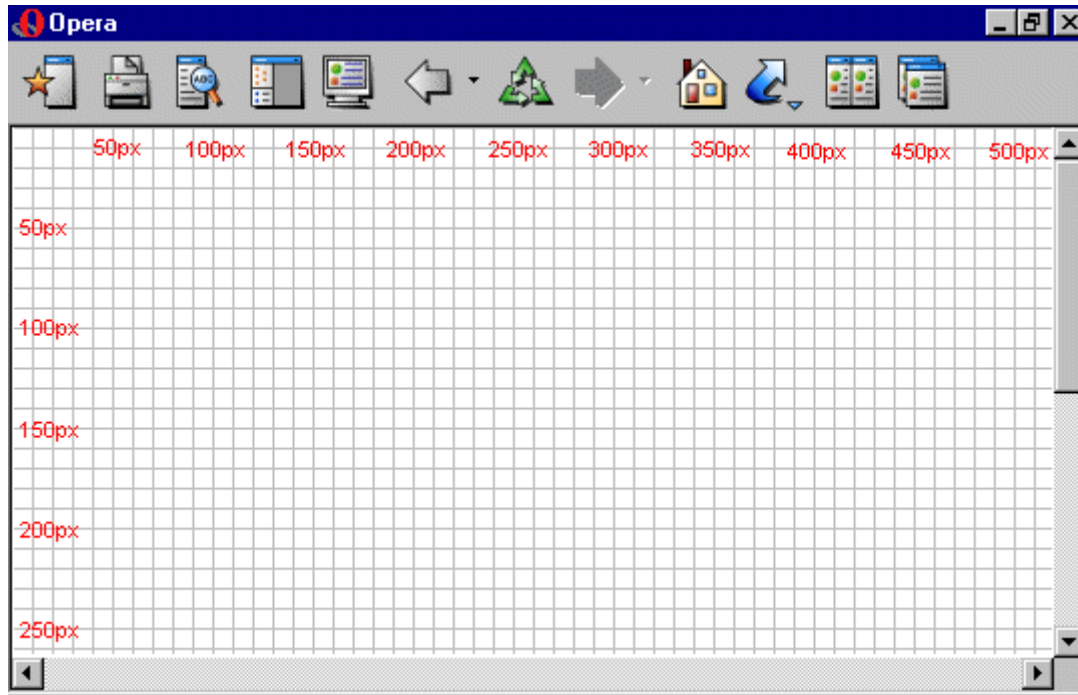
Dzięki pozycjonowaniu w CSS, możesz umieścić element dokładnie tam gdzie tylko chcesz. Razem z efektem pływania (patrz [lekcja 13](#)), pozycjonowanie daje ci nieograniczone możliwości tworzenia zaawansowanego i precyzyjnego układu na stronie.

Następujące rzeczy omówimy w tej lekcji:

- [Zasada pozycjonowania CSS](#)
- [Pozycjonowanie absolutne](#)
- [Pozycjonowanie relatywne](#)

## Zasada pozycjonowania CSS

Wyobraź sobie okno przeglądarki jako zbiór koordynatów:



Zasada pozycjonowania CSS mówi że możesz umieścić element gdziekolwiek chcesz w zbiorze koordynatów.

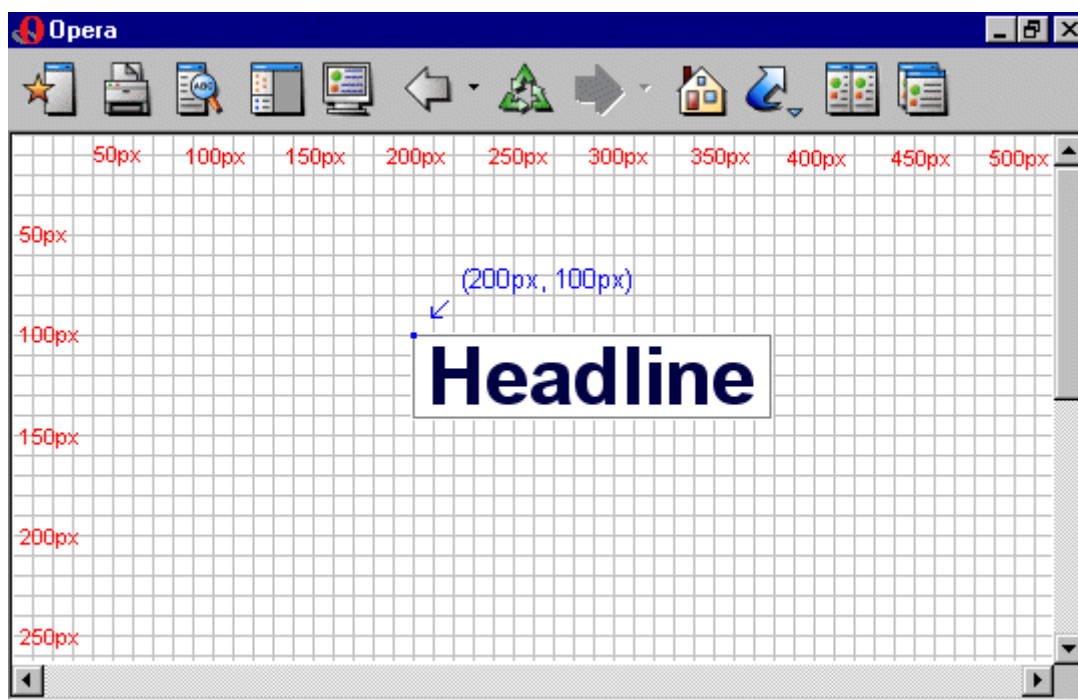
Powiedzmy że chcemy przenieść gdzieś nagłówek. Dzięki modelowi blokowemu (patrz [lekcja 9](#)) nagłówek pojawi się następująco:

**Headline**

Jeżeli chcesz umieścić nagłówek 100px od góry oraz 200px od lewej strony obramowania dokumentu, wykorzystaj następujący kod CSS:

```
h1 {
    position: absolute;
    top: 100px;
    left: 200px;
}
```

Rezultat będzie następujący :



Jak widzisz, pozycjonowanie CSS jest bardzo precyzyzną techniką układania elementów. Jest to o wiele łatwiejsze niż wykorzystywanie tabel, przeźroczystych obrazków lub czegokolwiek innego.

## Pozycjonowanie absolutne

Element pozycjonowany absolutnie nie zabiera miejsca w dokumencie. To oznacza że jego obecność nie zmienia ułożenia innych elementów.

Aby element pozycjonować absolutnie, właściwość `position` ustawiamy na wartość **absolute**. Możesz następnie użyć właściwości **left**, **right**, **top**, oraz **bottom** aby ustawić pozycję elementu.

Jako przykład pozycjonowania absolutnego, wybraliśmy 4 bloki, każdy w jednym rogu dokumentu:

```
#box1 {  
    position: absolute;  
    top: 50px;  
    left: 50px;  
}  
  
#box2 {  
    position: absolute;  
    top: 50px;  
    right: 50px;  
}  
  
#box3 {  
    position: absolute;  
    bottom: 50px;  
    right: 50px;
```

```
}  
  
#box4 {  
    position: absolute;  
    bottom: 50px;  
    left: 50px;  
}
```

- [Pokaż przykład](#)

## Pozycjonowanie relatywne

Aby pozycjonować element relatywnie, ustaw właściwość `position` na wartość **relative**. Różnica między pozycjonowaniem absolutnym i relatywnym jest taka, że pozycja jest kalkulowana w inny sposób.

Pozycja elementu który jest pozycjonowany relatywnie **obliczana jest na podstawie oryginalnej pozycji w dokumencie**. To oznacza, że przesuwasz element w prawo, lewo, dół lub w górę. Tym sposobem, element nadal zajmuje miejsce w dokumencie tam gdzie jest jego oryginalna pozycja.

Jako przykład takiego pozycjonowania, spróbujemy przedstawić trzy obrazki pozycjonowane relatywnie względem ich oryginalnego położenia. Zauważ jak elementy zostawiają puste miejsce w ich oryginalnym położeniu:

```
#dog1 {  
    position: relative;  
    left: 350px;  
    bottom: 150px;  
}  
#dog2 {  
    position: relative;  
    left: 150px;  
    bottom: 500px;  
}  
  
#dog3 {  
    position: relative;  
    left: 50px;  
    bottom: 700px;  
}
```

- [Pokaż przykład](#)

## Podsumowanie

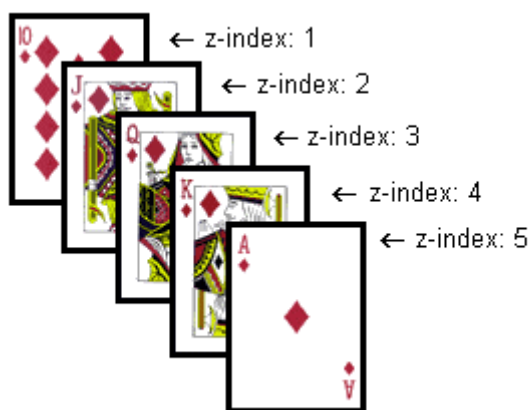
W poprzednich dwóch lekcjach, nauczyłeś się jak pozycjonować i ustawiać pływanie elementów. Te dwie metody dają ci wiele możliwości konstruowania twoich stron bez niepotrzebnego wykorzystywania metod w starym stylu (tabele, przeźroczyste obrazki). Zamiast tego wykorzystuj CSS. Jest bardziej precyzyjny, daje więcej wygody, i jest o wiele łatwiejszy w utrzymaniu porządku.

# Lekcja 15: Warstwy na warstwach a właściwość z-index

CSS operuje w trzech wymiarach - wysokości, szerokości oraz głębokości. Zapoznaliśmy się z dwoma. W tej lekcji, nauczymy się jak uczynić różne elementy warstwami. W skrócie, omówimy kolejność nakładających się na siebie elementów.

Do tego celu, każdemu elementowi możesz nadać numer (`z-index`). Reguła jest następująca: elementy z wyższym numerem nakładają się na elementy z niższym numerem.

Powiedzmy że gramy w pokera i mamy Pokera Królewskiego. Każda z kart w tym wypadku ma swój `z-index`:



W tym przypadku, liczby wykorzystaliśmy kolejne liczby (1-5) ale ten sam rezultat uzyskamy poprzez użycie 5 innych liczb. Najważniejsza jest tylko kolejność chronologiczna liczb (wielkość).

Kod w przykładzie z kartami może wyglądać następująco:

```
#ten_of_diamonds {
  position: absolute;
  left: 100px;
  top: 100px;
  z-index: 1;
}

#jack_of_diamonds {
  position: absolute;
  left: 115px;
  top: 115px;
  z-index: 2;
}

#queen_of_diamonds {
  position: absolute;
  left: 130px;
  top: 130px;
  z-index: 3;
}
```



```
}  
  
#king_of_diamonds {  
    position: absolute;  
    left: 145px;  
    top: 145px;  
    z-index: 4;  
}  
  
#ace_of_diamonds {  
    position: absolute;  
    left: 160px;  
    top: 160px;  
    z-index: 5;  
}
```

- [Pokaż przykład](#)

Metoda jest bardzo prosta ale możliwości jest wiele. Możesz umieścić obrazki na tekście, lub nad tekstem itd.

## Podsumowanie

Warstwy mogą zostać użyte w wielu sytuacjach. Na przykład, spróbuj użyć `z-index` do tworzenia efektów w nagłówkach zamiast wykorzystywania właściwości wraz z grafiką. Jeszcze jedna rzecz, pamiętaj że tekst ładuje się najszybciej, i może dostarczyć najwyższą rangę w wyszukiwarkach internetowych.

# Lekcja 16: Standardy WWW i sprawdzanie poprawności

W3C jest skrótem od [World Wide Web Consortium](#), co jest niezależną organizacją która zarządza standardem kodów w internecie (np. HTML, CSS, XML oraz inne). Microsoft, The Mozilla Foundation oraz inne są częścią W3C i zgadzają się co do przyszłego rozwoju standardów.

Jeżeli już trochę pracowałeś przy tworzeniu stron internetowych, prawdopodobnie wiesz już że mogą istnieć wielkie różnice w sposobie prezentacji strony zależnie od przeglądarki. Bardzo frustrujące i czasochłonne może być rozwiązywanie problemów związanych właśnie z tymi różnicami powstałymi w zależności jakiej przeglądarki używasz. Na przykład Mozilli, Internet Explorera, Opery lub całkiem innej przeglądarki.

Idea standardów pozwoli ułatwić pracę z technologiami internetowymi. To oznacza że obserwując standardy, programista internetowy zdaje sobie sprawę z odpowiedniego zachowania się jego stron niezależnie od platformy na której pracuje użytkownik. **Tak więc sugerujemy abyś sprawdził zgodność swojego kodu CSS ze standardem powszechnie stosowanym w internecie.**

## Walidator CSS

Aby ułatwić obserwację [standardów CSS](#), W3C stworzyło tak zwany [walidator](#) który czyta twój arkusz stylów i zwraca listę ostrzeżeń oraz błędów związany z twoim kodem CSS.

Aby ułatwić ci walidację arkusza stylów, możesz zrobić to prosto z tej strony. Po prostu zamień adres URL poniżej na adres twojego arkusza stylów i kliknij na przycisk. Zostaniesz poinformowany przez stronę W3C o wszystkich znalezionych błędach.

Kliknij tutaj aby sprawdzić CSS

Jeżeli walidator nie znajdzie błędów; pokaże się ten obrazek, który możesz umieścić na swojej stronie aby pokazać innym że twój kod jest zgodny ze standardami:



Dodatkowy walidator możesz znaleźć pod tym adresem: <http://jigsaw.w3.org/css-validator/>