# Dział: Kaskadowe arkusze stylów - CSS

# Temat: Budowa stylu

Kaskadowe arkusze stylów definiują jedynie sposób formatowania dokumentów HTML, ale ich nie tworzą. Elementy te muszą zostać zdefiniowane za pomocą znaczników w dokumencie HTML.

Ogólny schemat konstrukcji stylu wygląda tak:

**Selektor {**

**cecha: wartość;**

**cecha: wartość;**

**}**

Zamiast słowa ***selektor***podstawiamy znacznik z języka HTML, np.: znacznik akapitu <p>. Natomiast ***cecha*** jest wpisem ściśle określonym przez specyfikację kaskadowych arkuszy stylów. ***Wartość***  jest przypisana do danej cechy, ale też może być podobna do różnych cech. Praktyczny przykład stylu dla znacznika akapitu:

**p {**

**color: green;**

**font-size: 14px;**

**}**

Oczywiście można zapisać również tak:

p {color: green; font-size: 14px;}

cecha wartość

## Wstawianie stylów

Kaskadowe arkusze stylów definiują jedynie sposób formatowania dokumentów, ale ich nie tworzą.

W dokumencie HTML istnieje kilka sposobów na dołączanie stylów CSS:

1. Styl lokalny (w linii)
2. Wewnętrzny arkusz stylów
3. Zewnętrzny arkusz (w pliku)

**1 . Styl lokalny – w linii**

**Jest definiowany w tej samej linii, w której znajduje się element formatowany i umieszczany bezpośrednio w dokumencie HTML**

**Przykład 22, zawiera <body style=”background-color: #DFDFDF;”>**

**Znacznik <span> - rozciąganie stylu**

**<span> służy do grupowania kilku elementów strony w celu nadania im określonego stylu. Styl dla <span> ustala CSS**

**Przykład 23, zawiera:**

**<h3> Język <span style=”color: red”>CSS</span> określa układ graficzny dokumentu HTML</h3>**

**Znacznik <div> - wydzielone bloki**

**Za pomocą <div> elementy strony grupujemy w bloki i formatujemy za pomocą stylów. <div> to większy logiczny fragment strony.**

**Przykład 24, zawiera formatowanie bloku div**

**<div style=”width: 500px; height: 100px; background-color: Green;”>**

**2. Wewnętrzny arkusz stylów**

**Jest umieszczany w dokumencie HTML dzięki zastosowaniu znacznika <style>, który zawsze występuje w części nagłówkowej dokumentu <head>**

**Przykład 25**

**<style type=”text/css”>**

**p {**

**color: blue;**

**}**

**</style>**

**Przykład 26**

**3. Zewnętrzny arkusz stylów**

**Odnośnik umieszczony w dokumencie w sekcji <head> powinien mieć postać**

**<link rel=”stylesheet” type=”text/css” href=”nazwa\_pliku.css”>**

**href – ścieżka dostępu do pliku zawierającego style**

**Przykład 27**

**Import arkusza stylów**

**Polecenie importowania arkusza ma postać:**

**<style type=”text/css”>**

**/\* <! [CDATA[ \*/**

**@import url (adres importowanego arkusza.css);**

**/\* ] ] \*/**

**</style>**

**Plik z importowanym arkuszem może znajdować się w dowolnym miejscu (innym serwerze)**

## Kaskadowość arkuszy stylów

**Priorytety stylów (ważność):**

1. **Styl lokalny**
2. **Rozciąganie stylu**
3. **Wydzielone bloki, np. div**
4. **Wewnętrzny arkusz stylów**
5. **Zewnętrzny arkusz**
6. **Import stylów do zewnętrznego arkusza**
7. **Atrybuty definiowane w dokumencie HTML**

**Zasady kaskadowości można zmienić. Służy do tego polecenie !important umieszczone w stylu po wartości, której dotyczy. Użycie tego polecenia spowoduje, że właściwość zdefiniowana dla elementu będzie miała pierwszeństwo, nawet jeśli ma niższy priorytet.**

**Przykład 29**

## Selektory proste

Selektorem może być niemal dowolny znacznik HTML. Przykłady selektorów prostych:

|  |  |
| --- | --- |
| h1{  color: green;  font-size: 12pt;  } | li{  color: navy;  } |

……………………………………………………………………………………………………………………………………………………………

## Selektory uniwersalne:

Selektory proste odnoszą się jedynie do danego element – np. akapitu, i swoim działaniem obejmował tylko akapit, inne selektory(znaczniki) ignorował. Oczywiście można zdefiniować wiele selektorów prostych dla wszystkich elementów użytych na stronie WWW, jednak takie działanie nie ma większego sensu i prowadzi do niepotrzebnego zwiększania ilości kodu strony.

Jeżeli jakieś cechy i wartości są uniwersalne powinniśmy skorzystać z selektora uniwersalnego.

**\*{**

**color: navy;**

**}**

Selektor ma postać gwiazdki i działa dla wszystkich elementów strony. W tym przypadku dla tekstu na stronie: nagłówki, akapity, listy, tabele i ustawi kolor granatowy.

Innym selektorem uniwersalnym jest **BODY**, gdyż znacznik ten w hierarchii stoi najwyżej i jego właściwości będą dziedziczone przez inne elementy strony. Więcej o dziedziczeniu trochę później.

**body{ color: navy;}**

**Oba selektory: \* i body, zadziałają dokładnie tak samo i cały tekst w dokumencie będzie miał kolor granatowy.**

Jeżeli na twojej stronie pewne cechy są wspólne dla całego dokumentu, np. większość tekstu ma identyczny krój i kolor czcionki, to stwórz selektor uniwersalny i dodaj do niego wszystkie cechy i wartości, które powtarzają się na stronach.

## Selektor „potomka”

Przy użyciu selektora potomka można formatować elementy, które są zawarte wewnątrz innych znaczników, czyli leżą niżej w hierarchii drzewa dokumentu.

**H2{**

**fon-size: 20pt;**

**}**

A następnie chcemy, aby jakiś niewielki fragment tego nagłówka wyglądał inaczej, to musimy dopisać następujący kod do poprzedniej definicji:

**H2 { fon-size: 20pt;}**

**H2 i { fon-size: 10pt;}**

Kod HTML korzystający z selektora potomka będzie wyglądał następująco: [klik](sel_potomka.html)

Potomek nie musi być umieszczony bezpośrednio w znaczniku, którego jest potomkiem.

## Selektory „dziecka”

Selektory „dziecka” służy do definiowania formatowania elementów, które znajdują się o jeden rząd niżej w hierarchii drzewa dokumentu.

Ma postać:

Rodzic > dziecko

{

właściwość: wartość;

}

Gdzie symbol > oznacza bezpośredni związek między elementami. Znacznik będący dzieckiem musi być wewnątrz znacznika nadrzędnego.

Przykład selektora „dziecka”: [Plik](sel_dziecka.html)

## Selektory atrybutów

W języku CSS można formatować znaczniki na podstawie posiadanych atrybutów. Jeśli na przykład dla znacznika <p> zostanie zdefiniowany atrybut wyrównanie (align), to inaczej będzie wyświetlany akapit wyrównany do lewej, a inaczej wyśrodkowany.

Postać selektora atrybutów:

*selektor [atrybut= ”wartość”]*

*{*

*właściwość: wartość;*

*}*

*p [align= ”center”]*

*{*

*font-size: 16pt; color: silver;*

*}*

Przykład 2.19 – **prosty selektor atrybutu** jest wykorzystywany dla elementów o nadanym określonym atrybucie, którego wartość nie ma znaczenia.

**Selektor atrybutu o określonej wartości**

Postać:

*selektor[atrybut=”wartość”]*

*{*

*właściwość: wartość;*

*}*

*lub*

*[atrybut= ”wartość”]*

*{*

*właściwość: wartość;*

*}*

Przykład 2.20 – w tym przykładzie tylko pierwszy akapit został sformatowany zgodnie z definicją stylów, ponieważ tylko tu został zdefiniowany atrybut align z wartością center.

**Selektor atrybutu zawierającego określony wyraz**

Ten rodzaj selektora może zostać wykorzystany, gdy wartość atrybutu składa się z kilku wyrazów. Wystarczy że w wartości atrybutu wystąpi podany wyraz, aby element został odpowiednio sformatowany.

Postać selektora:

*selektor[atrybut~= ”wyraz”] {właściwość: wartość}*

*lub*

*[atrybut~= ”wyraz”] {właściwość: wartość;}*

Przykład 2.21

Zadanie do samodzielnego wykonania

Wykorzystując arkusze stylów oraz selektory atrybutów, zdefiniuj style, które pozwolą na wyświetlenie tekstów umieszczonych w akapicie w rożnych kolorach w zależności od wyrównania tych tekstów na stronie. Dla tekstów wyrównanych do lewej będzie to kolor niebieski, dla tekstów wyśrodkowanych – kolor zielony, a dla tekstów wyrównanych do prawej – kolor czerwony. W każdym przypadku czcionka o wielkości 18pt.

## Temat: Identyfikatory

Na stronie umieściliśmy kilka akapitów i wszystkie wyglądają tak samo, są koloru zielonego – poza jednym, który ma mieć tekst czerwony. Jak to zrobić?

**p{**

**color: green;**

**}**

**p#inny{**

**color: red;**

**}**

**p#pomponik{**

**color: orange;**

**}**

Pierwszy selektor ustawia kolor zielony dla wszystkich akapitów, drugi definiuje identyfikator, który wywołujemy:

**<p id=”inny”>Treść czerwonego akapitu</p>**

Oczywiście identyfikatory możemy tworzyć także dla innych selektorów prostych:

H1#maly{

font-size: 10pt;

} [plik](identyfikatory.html) [css](identyfi.css)

## Klasy

Klasa jest atrybutem znaczników html i odnosi się do selektora klasy lub klasy uniwersalnej.

Budowa klasy:

**Selektor.nazwa\_klasy**

**{**

**cecha: wartość;**

**}**

Praktyczny przykład klasy:

**p.zielona**

**{**

**color: zielony;**

**}**

Atrybut class dla dowolnego znacznika wygląda następująco:

**<znacznik class= ”nazwa\_klasy”>....</znacznik>**

**<p class=”zielona”>Zielony akapit</p>**

Tak zdefiniowana klasa może zostać wywołana tylko w znaczniku, dla którego została zdefiniowana. Dlatego jeśli ustawimy klasę dla znacznika akapitu, to nie możemy jej użyć dla nagłówka!!!

Wywołanie klasy w dokumencie umożliwia atrybut class – patrz wyżej. [plik](klasy.html), [css](klasy_arkusz.css)

Przykład 2.23

Podobnie jak w przypadku identyfikatora, możemy stworzyć klasę uniwersalną, którą można wywołać z dowolnym znacznikiem. Usuwamy nazwę selektora sprzed znaku kropki. Klasę uniwersalną możemy stosować do wszystkich znaczników w dokumencie HTML.

**.klasa\_uniwersalna**

**{**

**właściwość: wartość;**

**}**

Stwórzmy klasę uniwersalną wprowadzającą dla wybranych znaczników obramowanie zielone, kreskowane:

Listing CSS:

**.ramka**

**{**

**border: dashed green;**

**}**

Listing HTML wykorzystujący klasę uniwersalną: [plik](klasa_uniwersalna.html), oraz [css](klasa_uniwersalna.css)

[Właściwości CSS](#css)

**Zadanie do samodzielnego wykonania!!!**

**Zbuduj stronę, na której umieścisz trzy zdania.**

Ala ma ***zielonego*** kota.

**Adam** jedzie motorem.

Kot **poluje** na myszkę.

**Spraw aby:**

* *Przymiotnik – czerwony tekst, pochylony****,***
* **Rzeczownik – pomarańczowy, pogrubiony,**
* Czasownik – zielony, podkreślony

**W pierwszym zdaniu przymiotnik będzie na czerwono i pochylony**

**W drugim zdaniu rzeczownik jw.**

**W trzecim zdaniu czasownik jw.**

**Stronkę można urozmaicić wstawiając korespondującą z tekstem grafikę.**

## Pseudoklasy

Ogólnie – w CSS style są dodawane do elementów lub grup elementów na podstawie ich nazw, atrybutów lub zawartości. Można inaczej. Element może nabywać styl lub go tracić w związku z działaniem użytkownika lub zmieniać się w zależności od miejsca. Przykład – odsyłacze, które zmieniają swój wygląd po najechaniu na nie kursorem, kliknięciu lub otwarciu. Do takiej dynamicznej zmiany stylu elementu służą **pseudoklasy.**

Odsyłacze mają właściwości, które określają działanie użytkownika. Są to:

* :active – link odwiedzany, strona jest aktualnie wczytana;
* :link – link nieaktywny, nie został jeszcze odwiedzony;
* :visited – link odwiedzony, strona była otwierana;
* :hover – link gotowy do kliknięcia, kursor myszy nad linkiem.

Przy użyciu specjalnych selektorów dla odsyłaczy można definiować style, które będą nadawane odsyłaczom w zależności od ich bieżącego stanu.

**Odsyłacz podstawowy (pseudoklasa :link)**

a: link {właściwość: wartość;}

a:link {color: green; background: yellow;}

a.nazwa\_klasy: link{właściwość: wartość;}

a.nowa: link {color: orange;}

**Odsyłacz odwiedzony (pseudoklasa: visited)**

a: visited {właściwość: wartość;}

a.visited {color: red; background: silver;}

a.nazwa\_klasy: visited {właściwość: wartość;}

a.nowa2: visited {color: green;}

**Wskazanie kursorem myszy (pseudoklasa : hover)**

a: hover {właściwość: wartość;}

a.nazwa\_klasy: hover {właściwość: wartość;}

a.hover {color: yellow;}

a.nowa3: hover{color: orange;}

UWAGA! Pseudoklasa :hover może być używana do definiowania stylów elementów innych niż odsyłacze.

Przykład 2.29 pokazujący zastosowanie pseudoklasy :hover dla znacznika <div>

**Odsyłacz aktywny (pseudoklasa :active)**

Użytkownik nacisnął i trzyma przycisk myszki.

a: active {właściwość: wartość;}

a.active {color: blue;}

a.nazwa\_klasy: active{właściwość: wartość;}

a.nowa4: active {color: blue;}

**Kolejność deklarowania pseudoklas przypisanych do odsyłaczy ma znaczenie!!!**

Poprawna kolejność:

a:link {właściwość: wartość;}

a:visited {właściwość: wartość;}

a:hover {właściwość: wartość;}

a:active {właściwość: wartość;}

Przykład 2.31 – deklarowanie pseudoklas

**Pseudoklasa :fokus**

Nadaje atrybuty formatowania odsyłaczowi (wcześniej wybranemu) lub polu formularza , na którym został ustawiony kursor, na przykład gdy odsyłacz został wybrany klawiszem Tab lub w polu formularza znalazł się kursor.

Selektor :focus{właściwość: wartość;}

Zadanie do samodzielnego wykonania. Wykorzystując arkusze stylów, zdefiniuj pseudoklasy, które umożliwią dynamiczną zmianę stylu linków znajdujących się na Twojej stronie internetowej. Link podstawowy – kolor czerwony, wskazany przez myszkę – pomarańczowy, aktywny – kolor żółty. Strona ma dotyczyć kwalifikacji EE.09 i ma zawierać co najmniej takie informacje jak:

* Podstawa programowa kształcenia w zawodzie technik informatyk 351203 (jako dokument),
* Nazwa i opis kwalifikacji (strona główna), index.html
* Baner – pełna nazwa kwalifikacji
* Menu lewe – linki do podstron i plików
* Przykładowe pytania – egzamin pisemny (<http://www.kwalifikacjewzawodzie.pl> )
* Przykładowe zadanie praktyczne
* Link o nazwie : przydatne, gdzie umieścicie linki do stron z egzaminami on-line
* Stopka – autor strony

Strona ma być zbudowana na blokach div – szablony stron do pobrania mechanikryki.pl -> publikacje nauczycieli -> R.Dróbek -> Szablony stron na blokach <div>

W przypadku, gdy chcemy na stronie używać odnośników w różnych rozmiarach, musimy połączyć klasę z pseudoklasą lub identyfikatorem.

/\* Główna definicja domyślnych odsyłaczy\*/

**a : link**

**{**

**color: red;**

**font-size: 12pt;**

**}**

**a : visited**

**{**

**color: green;**

**font-size: 12pt;**

**}**

**a: hover**

**{**

**color: navy;**

**font-size: 12pt;**

**}**

**/\*definicja małych odsyłaczy\*/**

**a.maly : link**

**{**

**font-size: 8pt;**

**}**

**a.maly : visited**

**{**

**font-size: 8pt;**

**}**

**a.maly : hover**

**{**

**font-size: 8pt;**

**}**

**Style odpowiadające za wygląd mniejszych odsyłaczy zawierają jedynie wpisy odpowiedzialne za wygląd cech, które się zmieniają.**

W kodzie HTML – różnicowanie odsyłaczy:

**<a href=”http://helion.pl”>Domyślny, duży odnośnik</a>**

**<a href=”http://www.mechanikryki.pl” class=”maly”>mały odnośnik</a>**

# Grupowanie selektorów

**Podczas pracy ze stylami bardzo często bywa tak, że pewne wartości obejmują kilka elementów. Załóżmy, że wszystkie listy (wypunktowane, numerowane i definicji) mają mieć ten sam kolor. Zamiast definiować wpisy dla każdego oddzielnie, możemy zgrupować selektory:**

*ul, ol, dl*

*{*

*color: silver;*

*}*

*ul*

*{*

*color: red;*

*}*

Właściwości CSS

**Czcionka:**

|  |  |  |
| --- | --- | --- |
| Cecha | Opis | Możliwe wartości |
| font-family | Deklaracja kroju czcionki | - główne kroje czcionek: Arial, Verdana, Georgia, Times New Roman, Courier New, Helvetica, Tahoma;  - główne rodziny czcionek: serif, Sans-serf. |
| font-size | Deklaracja rozmiaru czcionki | Wysokość podawana w wybranej jednostce(1 cm, 10 pt, 15 px, 70%) lub jako predefiniowane wartości: xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger. |
| font-style | Deklaracja stylu czcionki | normal, italic, oblique |
| font-variant | Deklaracja typu czcionki | normal, small-caps |
| font-weight | Deklaracja grubości czcionki | lighter, normal, bold, bolder, 100, 200, 300, 400, 500, 600, 700, 800, 900 |
| font | Skrócona forma deklarowania wszystkich właściwości czcionki w jednym miejscu | Oddzielone spacjami wartości w następującej kolejności: style, variant, weight, size, family |

Przykład 1 – formatowanie akapitu **p**: krój czcionki – Courier New, styl – pochylenie, grubość – 500, rozmiar – 20 pixeli.

*p*

*{*

*font-family: „Courier New”;*

*font-style: italic;*

*font-weight: 500;*

*font-size: 20 px;*

*}*

**Tekst:**

|  |  |  |
| --- | --- | --- |
| Cecha | Opis | Możliwe wartości |
| color | Deklaracja koloru tekstu | Nazwy kolorów w RGB lub oznaczenia szesnastkowe |
| line-height | Odstęp między liniami tekstu | Wysokość w wybranej jednostce: 1 cm, 10 pt, 15 px, 70% |
| letter-spacing | Odstęp między znakami tekstu | Odstęp w wybranej jednostce |
| text-align | Poziome wyrównanie tekstu | left, right, center, justify |
| text-decoration | Wyróżnienie lub deklaracja czcionki | none, underline, overline, line-through, blink |
| text-indent | Wcięcie pierwszego wiersza | Długość w wybranej jednostce |
| text-transform | Zmiana wielkości liter w tekście | none, capitalize, uppercase, lowercase |
| word-spacing | Odległość pomiędzy wyrazami w tekście | Długość w wybranej jednostce |

Przykład 2 – formatowanie tytułu **H1**: dekoracja tekstu – nadkreślenie, odstępy między wyrazami – 10 px, wyśrodkowanie, kolor tekstu – zielony

*H1*

*{*

*text-decoration: overline;*

*word-spacing: 10 px;*

*text-align: center;*

*color: green;*

*}*

**Tło:**

|  |  |  |
| --- | --- | --- |
| Cecha | Opis | Możliwe wartości |
| background-attachment | Definiuje czy obrazek tła przewija się razem ze stroną, czy jest nieruchomy | fixed, scroll |
| background-color | Kolor tła | Kolor w RGB lub szesnastkowo |
| background-image | Obrazek jako tło strony | Adres URL |
| background-position | Pozycja obrazka tła | Współrzędne podawane w wybranych jednostkach (10 px 30 px), procentach lub wartości predefiniowane: top, left, right, bottom, center |
| background-repeat | Sposób powtarzania obrazka w tle strony | repeat, repeat-x, repeat-y, no-repeat |
| background | Skrócona forma deklarowania wszystkich właściwości tła w jednym miejscu | Oddzielone spacjami wartości w następującej kolejności: color, image, repeat, attachment, position |

Przykład 3 – ustawienie tła strony: tło obrazkowe (/grafika/tlo.jpg), nieruchome, bez powtarzania:

*body*

*{*

*background-image: url(/grafika/tlo.jpg);*

*background-attachment: fixed;*

*background-repeat: no-repeat;*

**}**

lub forma skrócona:

*body{ background: url(/grafika/tlo.jpg) no-repeat fixed}*

**Obramowanie:**

|  |  |  |
| --- | --- | --- |
| Cecha | Opis | Możliwe wartości |
| border-color  border-top-color  border-left-color  border-right-color  border-bottom-color | Deklaracja koloru wszystkich krawędzi lub wybranej krawędzi (górna, lewa, prawa, dolna) | Nazwa koloru w RGB lub zapis szesnastkowy koloru – domyślnie przeźroczyste |
| border-style | Styl wszystkich krawędzi lub wybranej gałęzi (górna, lewa, prawa, dolna) | none, hidden, dotted, dashed, solid, double, groove, ridge, inset, out set |
| border-width | Grubość wszystkich krawędzi lub wybranej krawędzi – jak wyżej (górna, lewa, prawa, dolna) | Grubość określana w pikselach lub wartość predefiniowana: thin, medium, thick |
| border | Skrócona forma deklarowania wszystkich właściwości obramowania w jednym miejscu lub wybranej krawędzi | Oddzielone spacjami wartości w kolejności: width, style, color |

Przykład 4 – ustawienie obramowania dla bloku **div:** styl – kropkowane, kolor – czerwony, grubość – średnia.

*div*

*{*

*border-style: dotted;*

*border-color: red;*

*border-width: medium;*

*}*

lub forma skrócona:

*div{border: medium dotted red}//tu skończono*

**Odsyłacze (linki):**

|  |  |
| --- | --- |
| Selektor:pseudoklasa | Opis |
| a:link | Deklaracja odsyłacza na stronie |
| a:visited | Deklaracja linku odwiedzonego |
| a:hover | Deklaracja linku po najechaniu myszką |
| a:active | Deklaracja linku przy kliknięciu |

Przykład 5 – zmiana wyglądu linków. Odsyłacz zwykły i odwiedzony: kolor – czarny, dekoracja – brak, tło – szare, obramowanie – linia ciągła. Odsyłacz aktywny i przy najechaniu myszką: kolor – srebrny, dekoracja – brak, tło – czarne, obramowanie – linia kreskowana.

*a:link, a:visited {*

*color: black;*

*text-decoration: none;*

*background-color: gray;*

*border-style: solid;*

*}*

*a:active, a:hover {*

*color: silver;*

*text-decoration: none;*

*background-color: black;*

*border-style: dashed***;**

**}**

**Listy:**

|  |  |  |
| --- | --- | --- |
| Cecha | Opis | Możliwe wartości |
| list-style-type | Rodzaj markera w listach wyliczeniowych | disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha, lower-greek, lower-latin, upper-latin, hebrew, armenian, georgian, cjk-ideographic, hiragana, … |
| list-style-position | Pozycja znacznika listy wypunktowanej | inside, outside |
| list-style-image | Deklaracja obrazka zastępującego marker w listach | Adres URL |
| list-style | Skrócona forma deklarowania wszystkich właściwości list | Oddzielone spacjami wartości w kolejności: type, position, image |

Przykład 6 – ustawienie obrazka *(plik zawierający grafikę, której chcemy użyć jako symbolu wypunktowania niech nazywa się np. gwiazdka.gif)* jako punktora graficznego w liście wypunktowanej **ol:**

*ol*

*{*

*list-style-image: url(gwiazdka.gif);*

*}*

**Kursor:**

|  |  |  |
| --- | --- | --- |
| Cecha | Opis | Możliwe wartości |
| cursor | Typ kursora | Adres URL pliku z kursorem lub właściwość predefiniowana: auto, crosshair, default, pointer, move, e-resize, ne-resize… |

Przykład 7 – zmiana wyglądu kursora na krzyżyk po najechaniu na tabelę:

*table, tr, td*

*{*

*cursor: crosshair;*

*}*

**Rozmiar:**

|  |  |  |
| --- | --- | --- |
| Cecha | Opis | Możliwe wartości |
| height | Wysokość elementu | Wysokość podawana w wybranej jednostce: 1 cm, 10 pt, 15 px, 20% |
| width | Szerokość elementu | Szerokość podawana w wybranej jednostce, jak wyżej |

Przykład 8 – ustawienie wysokości 95% i szerokości 80% dla elementu **div:**

*div*

*{*

*heihgt: 95%;*

*width: 80%;*

*}*

**Marginesy:**

|  |  |  |
| --- | --- | --- |
| Cecha | Opis | Możliwe wartości |
| margin-top  margin-left  margin-right  margin-bottom | górny  lewy  prawy  dolny margines | Odległość od krawędzi strony podawana w wybranej jednostce: cm, pt, px, % |
| margin | Forma skrócona deklarowania wszystkich marginesów jednocześnie | Wartości oddzielone spacjami w następującej kolejności: top, right, bottom, left;  Podanie jednej wartości spowoduje zmianę dla wszystkich marginesów;  Podanie dwóch wartości spowoduje odpowiednio zmianę dla górnego i dolnego marginesu, a następnie dla lewego i prawego;  Trzy wartości to zmiana dla marginesu górnego, prawego i lewego, dolnego (trzecia wartość) |

Przykład 9 – ustawienie marginesów dla strony: górny 0 px, prawy i lewy 10 px, dolny 5 px:

*body*

*{*

*margin: 0px 10px 5px;*

*}*

**Zadania do samodzielnego wykonania:**

1. Wprowadź następujący styl dla akapitu **p**:
   1. Kolor tekstu – czerwony,
   2. Krój czcionki – Arial,
   3. Rozmiar czcionki – średni,
   4. Tło akapitu – zielone,
   5. Obramowanie akapitu – kreskowane, pomarańczowe, 10 px
2. Wprowadź następujący styl dla bloku **div**:
   1. Kolo tekstu - biały,
   2. Krój czcionki – Verdana,
   3. Rozmiar czcionki – 18 px,
   4. Tło akapitu – szare,
   5. Obramowanie – podwójne, czarne,
   6. Odstępy pomiędzy liniami tekstu – 20 px,
   7. Kursor – punktowy,
   8. Marginesy – 10 px.

Oba Zadania możemy oczywiście rozwiązać w jednym arkuszu stylów. Posiłkując się swoimi notatkami należy na podpisanej kartce napisać listing arkusza stylów i oddać nauczycielowi. **Ściąganie od kolegi skutkuje oceną niedostateczną bez możliwości jej poprawy (zgodnie z PSO)**. Powodzenia!

Właściwości CSS

**Czcionka:**

|  |  |  |
| --- | --- | --- |
| Cecha | Opis | Możliwe wartości |
| font-family | Deklaracja kroju czcionki | - główne kroje czcionek: Arial, Verdana, Georgia, Times New Roman, Courier New, Helvetica, Tahoma;  - główne rodziny czcionek: serif, Sans-serf. |
| font-size | Deklaracja rozmiaru czcionki | Wysokość podawana w wybranej jednostce(1 cm, 10 pt, 15 px, 70%) lub jako predefiniowane wartości: xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger. |
| font-style | Deklaracja stylu czcionki | normal, italic, oblique |
| font-variant | Deklaracja typu czcionki | normal, small-caps |
| font-weight | Deklaracja grubości czcionki | lighter, normal, bold, bolder, 100, 200, 300, 400, 500, 600, 700, 800, 900 |
| font | Skrócona forma deklarowania wszystkich właściwości czcionki w jednym miejscu | Oddzielone spacjami wartości w następującej kolejności: style, variant, weight, size, family |

Przykład 1 – formatowanie akapitu **p**: krój czcionki – Courier New, styl – pochylenie, grubość – 500, rozmiar – 20 pixeli.

*p*

*{*

*font-family: „Courier New”;*

*font-style: italic;*

*font-weight: 500;*

*font-size: 20 px;*

*}*

**Przykład 2.35** – rodziny czionek

**Przykład 2.36** – rozmiar czcionki – 7 słów kluczowych w CSS

**Tekst:**

|  |  |  |
| --- | --- | --- |
| Cecha | Opis | Możliwe wartości |
| color | Deklaracja koloru tekstu | Nazwy kolorów w RGB lub oznaczenia szesnastkowe |
| line-height | Odstęp między liniami tekstu | Wysokość w wybranej jednostce: 1 cm, 10 pt, 15 px, 70% |
| letter-spacing | Odstęp między znakami tekstu | Odstęp w wybranej jednostce |
| text-align | Poziome wyrównanie tekstu | left, right, center, justify |
| text-decoration | Wyróżnienie lub deklaracja czcionki | none, underline, overline, line-through, blink |
| text-indent | Wcięcie pierwszego wiersza | Długość w wybranej jednostce |
| text-transform | Zmiana wielkości liter w tekście | none, capitalize, uppercase, lowercase |
| word-spacing | Odległość pomiędzy wyrazami w tekście | Długość w wybranej jednostce |

Przykład 2 – formatowanie tytułu **H1**: dekoracja tekstu – nadkreślenie, odstępy między wyrazami – 10 px, wyśrodkowanie, kolor tekstu – zielony

*H1*

*{*

*text-decoration: overline;*

*word-spacing: 10 px;*

*text-align: center;*

*color: green;*

*}*

**Tło:**

|  |  |  |
| --- | --- | --- |
| Cecha | Opis | Możliwe wartości |
| background-attachment | Definiuje czy obrazek tła przewija się razem ze stroną, czy jest nieruchomy | fixed, scroll |
| background-color | Kolor tła | Kolor w RGB lub szesnastkowo |
| background-image | Obrazek jako tło strony | Adres URL |
| background-position | Pozycja obrazka tła | Współrzędne podawane w wybranych jednostkach (10 px 30 px), procentach lub wartości predefiniowane: top, left, right, bottom, center |
| background-repeat | Sposób powtarzania obrazka w tle strony | repeat, repeat-x, repeat-y, no-repeat |
| background | Skrócona forma deklarowania wszystkich właściwości tła w jednym miejscu | Oddzielone spacjami wartości w następującej kolejności: color, image, repeat, attachment, position |

Przykład 3 – ustawienie tła strony: tło obrazkowe (/grafika/tlo.jpg), nieruchome, bez powtarzania:

*body*

*{*

*background-image: url(/grafika/tlo.jpg);*

*background-attachment: fixed;*

*background-repeat: no-repeat;*

**}**

lub forma skrócona:

*body{ background: url(/grafika/tlo.jpg) no-repeat fixed}*

**Obramowanie:**

|  |  |  |
| --- | --- | --- |
| Cecha | Opis | Możliwe wartości |
| border-color  border-top-color  border-left-color  border-right-color  border-bottom-color | Deklaracja koloru wszystkich krawędzi lub wybranej krawędzi (górna, lewa, prawa, dolna) | Nazwa koloru w RGB lub zapis szesnastkowy koloru – domyślnie przeźroczyste |
| border-style | Styl wszystkich krawędzi lub wybranej gałęzi (górna, lewa, prawa, dolna) | none, hidden, dotted, dashed, solid, double, groove, ridge, inset, out set |
| border-width | Grubość wszystkich krawędzi lub wybranej krawędzi – jak wyżej (górna, lewa, prawa, dolna) | Grubość określana w pikselach lub wartość predefiniowana: thin, medium, thick |
| border | Skrócona forma deklarowania wszystkich właściwości obramowania w jednym miejscu lub wybranej krawędzi | Oddzielone spacjami wartości w kolejności: width, style, color |

Przykład 4 – ustawienie obramowania dla bloku **div:** styl – kropkowane, kolor – czerwony, grubość – średnia.

*div*

*{*

*width:600px;*

*height: 600px;*

*border-style: dotted;*

*border-color: red;*

*border-width: medium;*

*}*

lub forma skrócona:

*div{border: medium dotted red}*

**Odsyłacze (linki):**

|  |  |
| --- | --- |
| Selektor:pseudoklasa | Opis |
| a:link | Deklaracja odsyłacza na stronie |
| a:visited | Deklaracja linku odwiedzonego |
| a:hover | Deklaracja linku po najechaniu myszką |
| a:active | Deklaracja linku przy kliknięciu |

Przykład 5 – zmiana wyglądu linków. Odsyłacz zwykły i odwiedzony: kolor – czarny, dekoracja – brak, tło – szare, obramowanie – linia ciągła. Odsyłacz aktywny i przy najechaniu myszką: kolor – srebrny, dekoracja – brak, tło – czarne, obramowanie – linia kreskowana.

*a:link, a:visited {*

*color: black;*

*text-decoration: none;*

*background-color: gray;*

*border-style: solid;*

*}*

*a:active, a:hover {*

*color: silver;*

*text-decoration: none;*

*background-color: black;border-style: dashed***;}**

**Listy:**

|  |  |  |
| --- | --- | --- |
| Cecha | Opis | Możliwe wartości |
| list-style-type | Rodzaj markera w listach wyliczeniowych | disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha, lower-greek, lower-latin, upper-latin, hebrew, armenian, georgian, cjk-ideographic, hiragana, … |
| list-style-position | Pozycja znacznika listy wypunktowanej | inside, outside |
| list-style-image | Deklaracja obrazka zastępującego marker w listach | Adres URL |
| list-style | Skrócona forma deklarowania wszystkich właściwości list | Oddzielone spacjami wartości w kolejności: type, position, image |

Przykład 6 – ustawienie obrazka *(plik zawierający grafikę, której chcemy użyć jako symbolu wypunktowania niech nazywa się np. gwiazdka.gif)* jako punktora graficznego w liście wypunktowanej **ol:**

*ol*

*{*

*list-style-image: url(gwiazdka.gif);*

*}*

**Kursor:**

|  |  |  |
| --- | --- | --- |
| Cecha | Opis | Możliwe wartości |
| cursor | Typ kursora | Adres URL pliku z kursorem lub właściwość predefiniowana: auto, crosshair, default, pointer, move, e-resize, ne-resize… |

Przykład 7 – zmiana wyglądu kursora na krzyżyk po najechaniu na tabelę:

*table, tr, td*

*{*

*cursor: crosshair;*

*}*

**Rozmiar:**

|  |  |  |
| --- | --- | --- |
| Cecha | Opis | Możliwe wartości |
| height | Wysokość elementu | Wysokość podawana w wybranej jednostce: 1 cm, 10 pt, 15 px, 20% |
| width | Szerokość elementu | Szerokość podawana w wybranej jednostce, jak wyżej |

Przykład 8 – ustawienie wysokości 95% i szerokości 80% dla elementu **div:**

*div*

*{*

*height: 95%;*

*width: 80%;*

*}*

**Marginesy:**

|  |  |  |
| --- | --- | --- |
| Cecha | Opis | Możliwe wartości |
| margin-top  margin-left  margin-right  margin-bottom | górny  lewy  prawy  dolny margines | Odległość od krawędzi strony podawana w wybranej jednostce: cm, pt, px, % |
| margin | Forma skrócona deklarowania wszystkich marginesów jednocześnie | Wartości oddzielone spacjami w następującej kolejności: top, right, bottom, left;  Podanie jednej wartości spowoduje zmianę dla wszystkich marginesów;  Podanie dwóch wartości spowoduje odpowiednio zmianę dla górnego i dolnego marginesu, a następnie dla lewego i prawego;  Trzy wartości to zmiana dla marginesu górnego, prawego i lewego, dolnego (trzecia wartość) |

Przykład 9 – ustawienie marginesów dla strony: górny 0 px, prawy i lewy 10 px, dolny 5 px:

*body*

*{*

*margin: 0px 10px 5px;*

*}*

**Zadania do samodzielnego wykonania:**

1. Wprowadź następujący styl dla akapitu **p**:
   1. Kolor tekstu – czerwony,
   2. Krój czcionki – Arial,
   3. Rozmiar czcionki – średni,
   4. Tło akapitu – zielone,
   5. Obramowanie akapitu – kreskowane, pomarańczowe, 10 px

**<p>Akapit formatowany w stylach zewnętrznych</p>**

1. Wprowadź następujący styl dla bloku **div**:
   1. Kolo tekstu - biały,
   2. Krój czcionki – Verdana,
   3. Rozmiar czcionki – 18 px,
   4. Tło akapitu – szare,
   5. Obramowanie – podwójne, czarne,
   6. Odstępy pomiędzy liniami tekstu – 20 px,
   7. Kursor – punktowy,
   8. Marginesy – 10 px.

**<div>Blok div formatowany w stylach zewnętrznych</div>**

Oba Zadania możemy oczywiście rozwiązać w jednym arkuszu stylów. Posiłkując się swoimi notatkami należy na podpisanej kartce napisać listing arkusza stylów i oddać nauczycielowi. **Ściąganie od kolegi skutkuje oceną niedostateczną bez możliwości jej poprawy (zgodnie z PSO)**. Powodzenia!

# Pozycjonowanie elementów

Pozycjonowanie odpowiada za umiejscowienie elementu w dowolnie wybranym miejscu przez zastosowanie właściwości *position.* Są dwa rodzaje pozycjonowania:

* Względne – *relative*
* Bezwzględne *– absolute*

Pozycjonowanie bezwzględne, nazywane absolutnym, pozwala na ustalenie położenia elementu w stosunku do ram okna przeglądarki lub innego elementu nadrzędnego. Wykorzystuje się w tym celu dodatkowo cztery właściwości: *left, right, top, bottom,* którym należy przypisać konkretne wartości w pikselach lub innych jednostkach miary, odpowiadające odległości od lewej, prawej, dolnej lub górnej krawędzi okna lub innego elementu.

*selector {*

*position: absolute;*

*left: wartość;*

*right: wartość;*

*top: wartość;*

*bottom: wartość;*

*}*

Przykład 1:

Pozycjonować możemy dowolny element na stronie. Jeśli chcemy pozycjonować obrazek (u mnie fotel.gif), należy mu nadać określoną nazwę klasy lub identyfikatora, a następnie wprowadzić odpowiednie właściwości w arkuszu stylów.

[css](fotel1.css), [plik](fotel1.html)

Obrazek otrzymał następujące właściwości w stylach:

* Obramowanie ciągłe, czarne;
* Pozycjonowanie absolutne w odległości 80 pikseli od lewej krawędzi okna;
* 40 pikseli od górnej krawędzi okna;
* Domyślne marginesy i dopełnienia (padding) ustawione na zero dla selektora body;
* Wprowadzone jest obrazkowe tło maui.jpg.

Standardowo wyświetlany element domyślnie posiada właściwość position o wartości *static*. Jeżeli zmienimy *static* na *absolute*, to staje się on elementem nadrzędnym dla znajdujących się w nim elementów. Pozycjonowane wewnątrz niego elementy będą ustawiane względem jego krawędzi.

Przykład 2:

W bloku *div* wstawimy obrazek, *div* będzie miał pozycjonowanie absolutne względem przeglądarki, a obrazek także, ale względem bloku div.

[css](fotel2.css), [plik](fotel2.html)

## Pozycjonowanie względne

Nazywane także relatywnym, pozwala na zmianę położenia elementu w stosunku do jego położenia początkowego. Wykorzystuje się w tym celu dodatkowo cztery właściwości (left, right, top, bottom), którym należy przypisać konkretne wartości odpowiadające przesunięciu w lewo, prawo, itd.

*selector*

*{*

*position: relative;*

*left: wartość;*

*right: wartość;*

*top: wartość;*

*bottom: wartość;*

*}*

Przykład 3:

Zbudujemy stronkę zawierającą tabelę składającą się z dwóch komórek. W pierwszej będzie tekst informujący, że tu nie ma pozycjonowania relatywnego, natomiast w drugiej komórce będzie obrazek, który jako jedyne formatowanie będzie miał ciągłe, czerwone obramowanie.

[css\_3](css_3.css), [plik\_3](plik_3.html)

Wstawiliśmy obrazek do komórki „normalnie”. Gdy wprowadzimy w arkuszu stylów dla obrazka pozycjonowanie relatywne (względne), zostanie on przesunięty o np. 30 pikseli od prawej krawędzi obrazka w położeniu początkowym(bez pozycjonowania), oraz o 30px od początkowego położenia od góry. Po przesunięciu obrazek przesłoni fragment tabeli. [css\_4](css_4.css), [plik\_4](plik_4.html)

Jeśli zmienimy położenie kilku elementów na stronie, może się zdarzyć, że będą one na siebie nachodzić, a nawet przesłaniać się wzajemnie. Warto wtedy ustalić, w jakiej kolejności elementy będą się na siebie nakładać. Służy do tego właściwość ***z-index***, przyjmująca wartości liczbowe od 1. Jedynka oznacza element na samym spodzie, natomiast element o największej wartości z-index będzie na wierzchu lub jak kto woli, będzie zakrywał pozostałe „pod nim”.

Przykład 5:

Wyświetlimy ten sam obrazek 2 razy ale kaskadowo. Każdy ma własny identyfikator. Na samym spodzie obrazek ma identyfikator obrazek\_css oraz czarne ciągłe obramowanie. Na nim obrazek posiadający identyfikator obrazek2\_css oraz obramowanie czerwone, kreskowane. Na samym wierzchu umieścimy napis JAKIŚ TEKST w kolorze niebieskim. Odstępy od krawędzi okna w arkuszu.

[css\_5](css_5.css), [plik\_5](plik_5.html)

Korzystając z listingu 1.1 i listingu 1.2, utwórz czerwony kwadracik.

Listing 1.1 (html)

**<div class = ”czerwony” >czerwony kwadrat</div>**

Listing 1.2 (css)

***div.czerwony***

***{***

***border-style: solid;***

***background-color: red;***

***width: 200px;***

***height: 200px;***

*}*

**Zadanie domowe 3** - Wykorzystując dowolny rodzaj pozycjonowania, umieść utworzony obrazek w następujących układach:

1. Umieść kwadracik w lewym górnym rogu przeglądarki, 100px od lewej i 100px od góry.

|  |
| --- |
| Czerwony |

1. Wprowadź cztery dodatkowe kwadraciki w odpowiednich kolorach i wykonaj poniższy układ:

|  |  |  |
| --- | --- | --- |
| Niebieski |  | Żółty |
|  | Czerwony |  |
| Zielony |  | Srebrny |

1. Zmień rozmiar kwadratów i umieść jeden na drugim w takiej kolejności:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| srebrny | | | | | | | | |
|  | żółty |  | | | | | |  |
| zielony | | | | |  |
|  |  | niebieski | | |  |
|  | czerwony |  |
|  | |
|  | | | |
|  | | | | | |
|  | | | | | | | |

Oba Zadania możemy oczywiście rozwiązać w jednym arkuszu stylów. Posiłkując się swoimi notatkami należy na podpisanej kartce napisać listing arkusza stylów i oddać nauczycielowi. **Ściąganie od kolegi skutkuje oceną niedostateczną bez możliwości jej poprawy (zgodnie z PSO)**. Powodzenia!

**RAMKI**

Ramki odpowiadają za podział obszaru strony na charakterystyczne bloki. Każdy z nich może wyświetlać inną stronę internetową lub inny obraz.

**Cechą charakterystyczną strony indeksowej opisującej podział strony na ramki jest brak znacznika <body>. Zastępuje go znacznik <frameset> …. </frameset>. Stronę można podzielić na kolumny – cols i wiersze – rows.**

**Podstrony ładowane do poszczególnych ramek mają body.**

[**L131**](ramki/pionowo.html)

**Podział strony na trzy kolumny:**

<frameset cols=”100, 140, \*”>

<frame src=”strona1.html”>

<frame src=”strona2.html”>

<frame src=”strona3.html”>

<noframes>

<body>

Widok dla przeglądarek nieobsługujących ramek

</body>

</noframes>

</frameset>

**L132**

**Podział strony na dwa wiersze:**

<frameset rows=”30%, 30%, \*”>

<frame src=”strona1.html”>

<frame src=”strona2.html”>

<frame src=”strona3.html”>

<noframes>

<body>

Widok dla przeglądarek nieobsługujących ramek

</body>

</noframes>

</frameset>

Znacznik **frame** może mieć kilka atrybutów:

* **noresize –** blokuje zmianę rozmiaru ramki;
* **frameborder –** 1 – obramowanie widoczne, 0- niewidoczne
* **scrolling** – yes- suwak widoczny, no – niewidoczny, auto – pojawia się gdy potrzeba
* **marginwidth** – margines poziomy w pikselach
* **marginheight** – pionowy w pikselach
* **name** – nazwa ramki

**L133**

**Baner, menu lewe, główna**

<frameset rows=”150, \*”>

<frame src=”logo.html”>

<frameset cols=”25%, \*”>

<frame src=”menu.html”>

<frame src=”start.html” name=”ramkaglowna”>

</frameset>

<noframes>

<body>

Widok dla przeglądarek nieobsługujących ramek

</body>

</noframes>

</frameset>

Sprawdź zwoje umiejętności

Wykonaj stronę opartą na poniższym układzie ramek. W każdej ramce wczytaj inną stronę internetową. Pierwsza ramka: 30% szerokość, 50% wysokość, pozostałe ramki analogicznie. Przy tworzeniu pliku wybierz Plik –Nowy -HTML 4.01 Frameset (HTML 5 nie obsługuje ramek)

|  |  |  |
| --- | --- | --- |
| 30%, 50% | \*, \* | 30%, \* |
| 30%, 50% |
| 30%, 50% |
| 30%, \* |