

# Zadanie obrazkowe w HTML – rozwiązanie

dr Ryszard Wiśniowski

23 marca 2020

## 1 Zadanie

Stworzyć stronę w HTML na której znajdzie się 12 obrazków, rozmieszczonych po 3 w każdym wierszu.

## 2 Założenia projektu

### 2.1 Co musimy mieć, czyli wymagania klienta

1. HTML 5
2. 12 obrazków w układzie  $3 \times 4$ .

### 2.2 Co chcemy mieć, czyli elastyczność projektu

Na tym etapie warto przemyśleć, co nasz klient może chcieć zmienić. Najlepiej tak zaprojektować stronę, aby te zachcianki dało się szybko zrealizować. Naszą potencjalną zachcianką będzie zmiana układu obrazków np na  $4 \times 3$

## 3 Epoka kamienia łupanego – tak nie robimy.

Dawno, dawno temu, do tworzenia układu elementów strony używano tabel. Wyglądało to tak:

```
<table>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
... kolejne wiersze podobnie ...
</table>
```

To najgorsze możliwe rozwiązanie, bo aby zmienić układ musimy przebudować cały kod strony. W dodatku mnóstwo znaczników i łatwo coś popsuć

## 4 Kamień łupany 2.0

Można też tak:

```
<div>



</div>
... kolejne wiersze podobnie ...
```

Czytelniejszy kod, ale gdy obrazki nie zmieszczą się po 3 w jednej linii, to cały układ się rozleci na kawałki. Dalej zmiana układu wymaga grzebania w HTML-u, ale już w mniejszym stopniu.

## 5 XXI wiek – jedna z propozycji

W dzisiejszych czasach za wygląd strony odpowiada CSS. Daje on kilka sposobów rozwiązania naszego zadania, ja przedstawię jedno z nich oparte o tzw. **grid**, czyli taki uwspółcześniony odpowiednik kartki w kratkę. Metoda ta jest często używana na stronach, których wygląd przypomina układ „gazetowy”. (Ta technika zresztą została „ściągnięta” od redaktorów gazet).

### 5.1 plik HTML

```
<!doctype html>
<head>
<meta charset="utf-8"/>
<link rel="stylesheet" href="styl.css"/>
</head>
</body>
<div class="ramka">


...

</div>
</body>
</html>
```

W sekcji **head** znacznik **link** połączy nam ten dokument z plikiem *styl.css*, który będzie całkowicie odpowiadał za stronę „artystyczną”. Zamiast takiego rozwiązania, można też całą zawartość tego pliku umieścić wewnątrz w sekcji **style**:

```
<style>
... tu zawartość pliku styl.css
</style>
```

Która **koniecznie** musi się znaleźć wewnątrz sekcji **head** W HTML-u nie ma nawet informacji o układzie tych obrazków. Aby można było rozbudowywać tę stronę o inne elementy, całą galerię umieściłem w osobnym bloku **div** i aby można mu było nadawać specyficzne właściwości nadałem mu klasę „ramka”. (klasa oznacza zbiór właściwości)

### 5.2 Plik styl.css

```
.ramka{
display: grid;
grid-template-columns: auto auto auto;
width:90%;
margin:auto;
}
.ramka>img{
width:300px;
height:200px;
display: block;
border:solid black 1px;
margin:5px;
}
```

Najpierw deklarujemy właściwości klasy „ramka”, czyli naszego kontenera galerii. Aby zaznaczyć, że chodzi nam o klasę a nie nazwę znacznika, przed jej nazwą stawiamy kropkę. I tak po kolei:

- zawartość kontenera ma być wyświetlana w układzie siatki (czyli takiej wirtualnej tabelce)
- szablon tej siatki to 3 kolumny o takich samych szerokościach, w zasadzie dowolnych, byle by były równe. (teraz widać, jak prosto zmienić ten układ na 4 kolumny). Wysokość wierszy sama się dostosuje (to działa podobnie jak komórki w Excel-u).
- szerokość tego kontenera to 90% szerokości bloku, który go zawiera (czyli u nas całej strony)
- automatyczne marginesy. To taki trik, aby wycentrować ten element.

Kolejna sekcja to obrazki. Selektor „.ramka>img” oznacza znacznik `img` znajdujący się bezpośrednio w obiekcie klasy „.ramka”. Pierwsze dwa wiersze służą do ustalenia wymiarów obrazków. Wszystkie będą takie same i będzie to ładnie wyglądało. Kolejne linie oznaczają odpowiednio:

- obrazek ma być traktowany jak blok (czyli element „przesuwalny”), a nie jak wyraz (domyślnie) czyli coś, czego położenie wynika z szyku zdania. Jak obrazki się „buntują” nie chcą się ustawiać tam, gdzie chcemy, to ten zapis załatwia sprawę.
- obramowanie ciągłą linią czarną o grubości 1px. Nie musi być konieczne, ale odgradza obrazek od tła i lepiej to wygląda.
- odstęp od sąsiadów, aby jeden nie sklejał się z drugim.

To taki szkielet, który można udoskonalać do woli.

## 6 Inne sposoby

Można użyć np. technologii **flex** zamiast **grid**. Jej zaletą jest to, że ilość obrazków w jednym wierszu dostosowuje się do wymiarów okna. Klient (czyli ja) tego nie chciał, ale może się skusi :). Polecam stronę <https://www.w3schools.com/> Jest tam pełno przykładów obu rozwiązań. Zresztą z napływających prac wynika, że tam zagłądacie.

## 7 Błędy i wypatrzenia

Nie robi błędów ten, który nic nie robi, więc nie przyjmujemy się tym, że coś się nie dało. Generalnie bardzo dobrze poradziście sobie z tym zadaniem.

Najczęstsze błędy, które zauważyłem:

- Sekcja **style** poza sekcją **head**. Jedyne słuszne miejsce to **head**. Nie można tej sekcji umieszczać wewnątrz **body**, ani pomiędzy sekcjami.
- Niedomknięte znaczniki. Każdy otwarty znacznik musi być domknięty. Znaczniki blokowe domykamy tak: `<znacznik>...</znacznik>` a punktowe (takie jak np. **img**), tak: `<znacznik />`. Nie wolno też przecinać znaczników, to znaczy wszystkie znaczniki, otwarte wewnątrz jakiegoś bloku, muszą być zamknięte przed zamknięciem tego bloku.
- Przeszarzałe rozwiązania. Znaczniki `<center>`, `<i>`, `<b>` oraz atrybuty wymiarów `width`, `height` - mimo, że działają - nie powinny być stosowane. Oficjalnie są uznawane za przestarzałe, a to oznacza, że ich godziny są policzone. Zamiast tego stosuje się odpowiednie zapisy w CSS.
- Style lokalne, czyli atrybut `style="..."`. Należy go używać tylko wtedy, gdy nie ma innej możliwości (Mniej więcej tak często, jak broń nuklearną). Lepiej utworzyć sobie osobną klasę w CSS. Dzięki temu hurtowo zarządzamy wszystkim obiektami tej klasy z jednego miejsca, a poza tym mamy mniej pisania.

To nadużycie (bo błąd to nie jest) występowało w waszych pracach najczęściej, prawdopodobnie dlatego, że w tutorialach w3schools jest ono często stosowane, aby nie odwracać uwagi od celu przykładu.

## 8 Podsumowanie

Zadanie wykonane bardzo dobrze, i tego się trzymajmy. Teraz spróbujcie, bazując na tym doświadczeniu, zrobić pierwsze zadanie projektowe, czyli blog kulinarny. Chwilowo nie mogę Wam wpisywać ocen (mamy zakaz), więc nie oceniam, ale doceniam :)

Milego dnia

Ryszard Wiśniowski