

lista zadań na 19-25.listopada

[link do tutoriala](#)

To już przedostatni tydzień! Mam nadzieję, że zdążyłaś/eś już złapać koderskiego bakcyła. Wyzwanie jest dostępne do końca listopada, więc jeśli ktoś z Twoich znajomych chce dołączyć, to zapraszam! Ostatni dzwonek!

W tym tygodniu przyszła pora na zapoznanie się z pozycjonowaniem elementów, czyli właściwością CSS, jaką jest `position`.

Dzięki pozycjonowaniu m.in. nawigacja będzie mogła zostać przyklejona do góry strony.

zadanie 1.

Przeczytaj rozdział 8.3 z tutoriala. Dzięki niemu poznasz teorię i będziesz gotowa/y, żeby zabrać się za wypożycjonowanie nawigacji.

W projekcie założyłam, że nawigacja będzie przyklejona do górnej części strony. Możemy to zrealizować dzięki `position: fixed;`

Przypisz tę właściwość Twojej nawigacji w CSS-ie.

Po dodaniu jej do CSS-owej klasy/selekторa (w moim przypadku `nav { }`) nawigacja wskoczy nam na inne miejsce niż zaplanowałaś/eś. Kolejny krok, to dodanie odpowiednich wartości pozycji. W tym przypadku eksperymentuj z wartością dla `top`. Sprawdź, co stanie się gdy wpiszesz: `top: 0;`

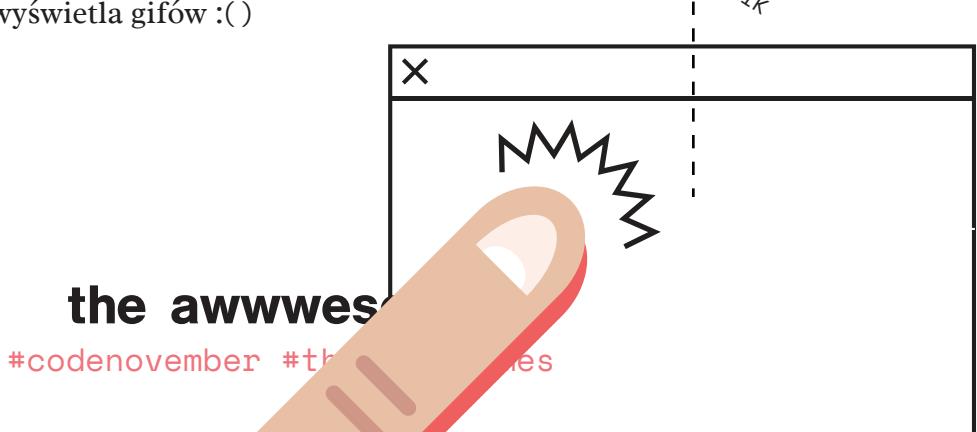
Ważne będzie jeszcze wsadzenie nawigacji do „wyższej warstwy”, tak, żeby przy scrollowaniu zawsze była na wierzchu Wykorzystaj do tego właściwość `z-index`. `z-index` decyduje, na której warstwie znajduje się dany element – czy ma być on na wierzchu czy pod spodem w relacji do pozostałych elementów. Jeśli korzystałaś/eś kiedyś z Photoshopa to `z-index` działa na tej samej zasadzie co warstwy.

Wartość `z-index` podajemy bez żadnych jednostek. Im większa wartość, tym w wyższej warstwie ląduje element. Na starcie wszystkie elementy mają domyślną wartość `z-index: 0`. Możesz stosować również wartości ujemne. Wystarczy, że przypiszesz nawigacji `z-index: 1`, żeby zawsze była na wierzchu innych elementów.

Ważne! `z-index` działa jedynie na elementy, które mają przypisane `position: absolute`, `relative` lub `fixed`. Domyślnie każdy element posiada `position: static`.

Efekt, który powinnaś/powinieneś osiągnąć przedstawia następujący gif --- (szkoda, że format .pdf nie wyświetla gifów :()

KLIK
KLIK



zadanie 2.

Pora na przyklejenie ikonek social media do lewej krawędzi strony.

Działanie jest analogiczne jak przy zadaniu
z przyklejaniem nawigacji.



Ikonek nie dodawałam wcześniej w kodzie (nie znajdziesz
ich w pliku z rozwiązaniem), więc trzeba to nadrobić. ☺

Proponuję zaraz po sekcji <header> dodać
w kodzie HTML sekcję, w której znajdą się ikonki.

Ikonki powinny się ustawić jedna pod drugą – możesz użyć
do tego znacznika listy . Oczywiście inne pomysły też są mile widziane!

Nie przeraź się, jeśli odświeżysz stronę – ikonki będą olbrzymie.

Dzieje się tak, gdyż są to pliki w formacie SVG, czyli w formacie
grafiki wektorowej. Grafika wektorowa to zazwyczaj kształty i linie, które
moga rozciągać się w nieskończoność (nie mają zdefiniowanego rozmiaru) i nie tracą na
jakości #win. Żeby je okiełznać będziesz musiał/a dodać właściwość height lub width.
Wystarczy, że zdefiniujesz jedną z nich, a druga dostosuje się automatycznie, tak by
zachować proporcje grafiki.

zadanie 3.

W tym zadaniu nauczysz się jak pozycjonować elementy
względem jego rodzica, czyli kontenera, w którym się
znajduje dany element. #trochęMasłoMaślane, ale zostań ze mną.
Do takiego pozycjonowania wykorzystamy
position: relative oraz position: absolute.
Twoje zadanie polega na ułożeniu napisu „layout by The Awwwe-
somes” (ważne! ☺) w lewym dolnym rogu sekcji <header>.



Na razie nie przejmuj się tym, że napis jest przekręcony o 90 stopni, będziemy się
nad tym „pastwić” już za tydzień.

Relacja rodziniec-dziecko przedstawia się następująco:

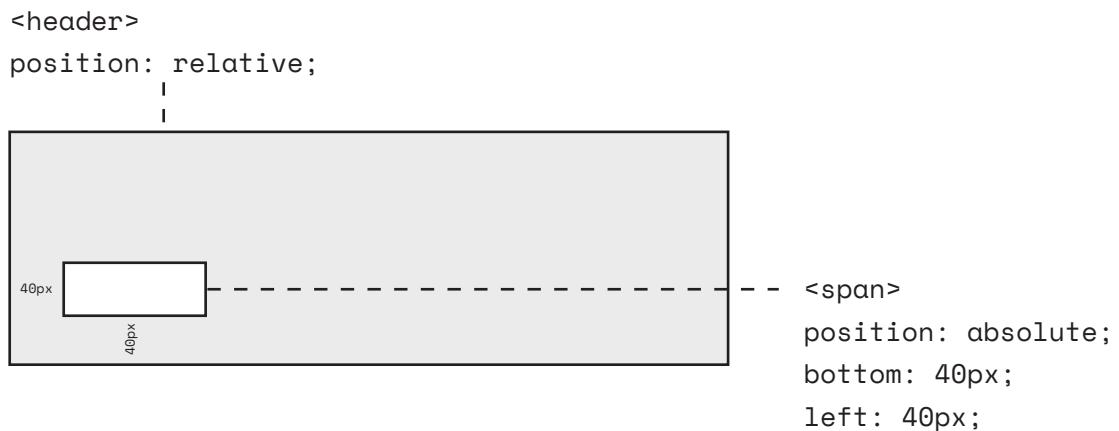
<header> jest naszym rodzicem, w którym znajduje się dziecko, czyli napis
„layout by...” (może być to zwykły <p> lub).

Rodzic powinien mieć position: relative, względem której będzie
pozycjonowanie dziecko. Dziecko będzie mieć position: absolute.

Ja zapamiętałam to w ten sposób: rodzic zaczyna się na literę R, więc musi mieć
pozycję relative, a dziecko to dziecko, nie ma nic do gadania i zostaje

tzw. absolutem #nieOceniajMnie, przynajmniej to jakoś wbiło mi się w głowę ☺

Na następnej stronie znajdziesz schemat.



Nadając elementowi wypożyczonowanemu absolutnie wartość top: 0; oraz left: 0; ustawi się on w lewym górnym rogu najbliższego elementu, który ma pozycję relative (w naszym przypadku odbędzie się to względem header'a).

Jeśli żaden z elementów nie będziecie mieć zdefiniowanej position: relative, to element będzie wypozycjonowany względem krawędzi całej strony, czyli <body>.

Jeśli w tej sytuacji nadasz elementowi bottom: 0, to pojawi się on na samym dole strony.

zadanie 4.

Przeczytaj cały rozdział 9. poświęcony złożonym selektorom CSS.

Skup się na pseudoklasach i pseudoelementach.

Pseudoelement wykorzystamy do zakodowania obrazka znajdującego się przy zdjęciu, a także elementu dekoracyjnego, czyli pomarańczowego „maziaja” przy nagłówku „let's stay in touch”.



Musisz stworzyć pseudoelement `::after` dla kontenera, w którym znajduje się zdjęcie z panią. Obrazek pomarańczowego x dodaj w `content: "url(...)"`; gdzie w `url` umieścisz odpowiednią ścieżkę do obrazka `cross.svg` (tak jak zrobiłaś/eś to kiedyś dla tła w sekcji `header`, pamiętasz?). Następnie zdefiniuj `position: absolute` (bedziemy pozycjonować obrazek względem

Następnie zaznaj, pozycję - **absolute** (szczególnie pozyując obrazek w względem kontenera ze zdjęciem). To teraz pytanie: jakie position powinien mieć ten kontener? Odpowiedź znajdziesz w poprzednim zadaniu. :-)

Odpowiedz znajdziesz w poprzednim zadaniu \ominus
Bo za tym potrzebujemy jeszcze nadejścia szeroki

Poza tym potrzebujemy jeszcze nadać jakąś szerokość (widżet) pseudoelementowi, żeby obrazek został wygenerowany.

Nasz pseudoelement ma się wyświetlić pod zdjęciem (pod spodem), więc będzie miał `z-index` o mniejszej wartości, czyli np. `z-index: -1;`
Do odpowiedniego wypożycjonowania obrazka śmiało używaj `left` i `top`, pamiętaj, że możesz używać ujemnych wartości do odsuwania obrazka od krawędzi kontenera rodzica.

Pseudoelementy mogą wydawać się trochę skomplikowane – jest tyle właściwości do dodania, żeby zadziałyły tak jak trzeba! Ale nie zniechęcaj się. Poniżej umieszczam przykład dla jaśniejszego przedstawienia sprawy.

przykład

```
h2.decorated {  
    position: relative;  
    ...  
}  
  
h2.decorated::before {  
    content: "url(ściezka do obrazka)";  
    position: absolute;  
    width: 800px;  
    z-index: -1  
    top: 0;  
    left: -100px;  
}
```

pseudoelement

To tyle w tym tygodniu!

Nauczyłaś/eś się jak przyklejać elementy, jak pozycjonować je względem siebie używając relacji rodzic-dziecko (`position: relative`, `absolute`) oraz jak korzystać z pseudoelementów .

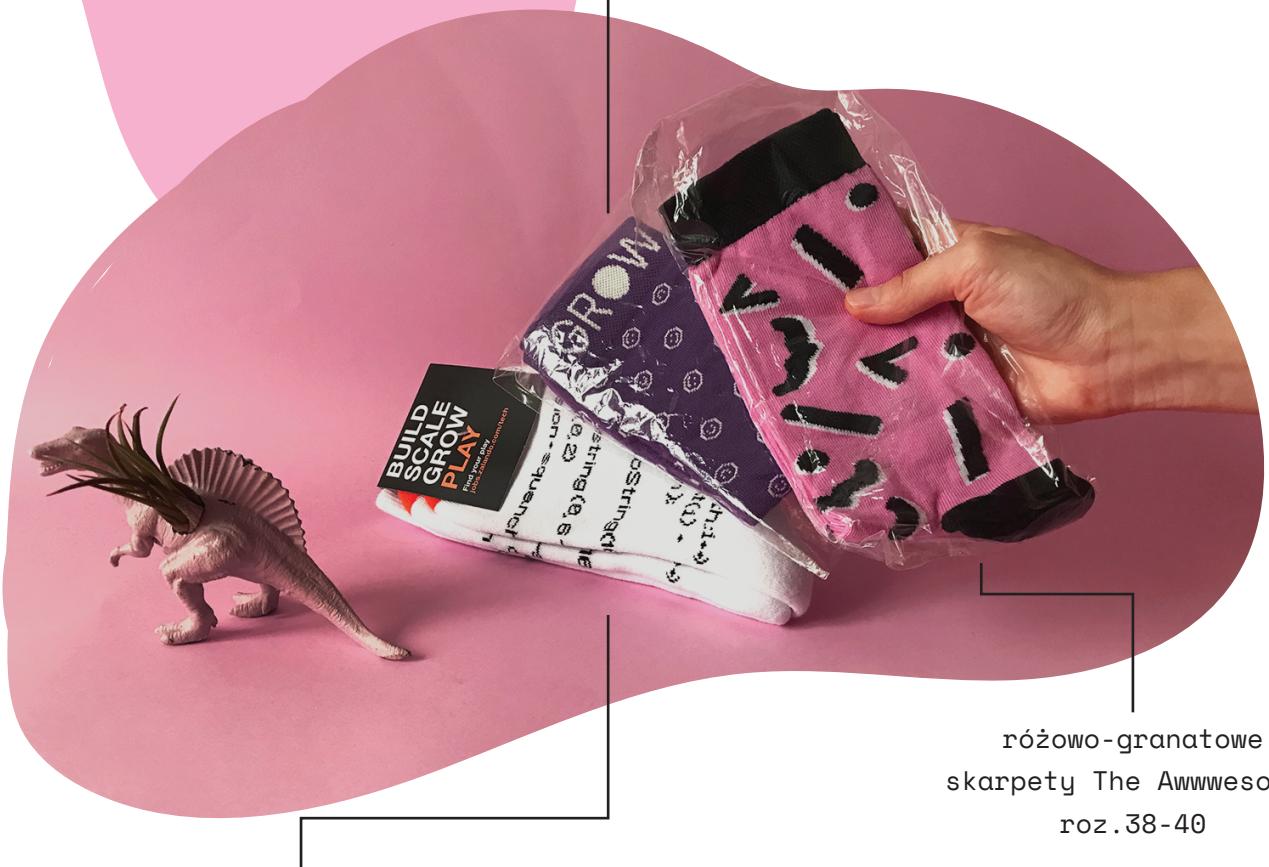
Kolejna lista zadań, którą otrzymasz będzie... ostatnią! Dasz wiarę?! Przy okazji jestem mega ciekawa czy uważasz, że dzięki wyzwaniu czujesz się pewniej z kodowaniem? Masz z tego frajdę? A może coś Cię frustruje? Daj znać!

A teraz jeszcze chciałabym ogłosić **KONKURS**, który zapowiadałam już na Instagramie @theawwesomes w stories.

Jeśli chcesz zdobyć jedną z trzech, absolutnie wyjątkowych, koderskich par skarpetek (do zweryfikowania na następnej stronie) lub naklejki The Awwwesomes, to udostępnij w social media, jak idzie Twoje wyzwanie. Nie zapomnij otagować @theawwesomes.
Spośród wszystkich wylosuję zwycięzców :)

the awwesomes

#codenovember #theawwesomes



fioletowe
skarpety GROW
roz. 35-37

skarpety Zalando
roz. 36-41
(mam też męskie 41-46)

różowo-granatowe
skarpety The Awwwesomes
roz. 38-40

+ naklejeczki

Konkurs czas start!

Macie czas do 25.II (czyli do przyszłej niedzieli włącznie!).

Potem wylosuję zwycięzców, z którymi się skontaktuję i roześlę nagrody.

Losowanie udostępnię na Instagram stories.

Trzymam kciuki, piąteczka!

- Aga ☺

the awwwesomes

#codenovember #theawwwesomes