Lekcja 2 - podsumowanie

Na zajęciach zauważyliśmy, że pliki html, możemy otwierać z pulpitu. Html jest interpretowany tylko przez przeglądarkę. Inaczej sytuacja wygląda z plikami php, do przetworzenia ich potrzebny jest serwer które będzie je interpretował i wyślę nam wyniki swoich obliczeń.

**Serwer** - jest to komputer w sieci internetowej który ma zainstalowane specjalne usługi umożliwiające udostępnianie plików do internetu. W systemach *Linux* usługa ta nazywa się *apache2*, w przypadku systemów *Windows* jest to *IIS*.

Na zajęciach do serwowania plików użyliśmy aplikacji *Xampp*, która załatwiła nam całą trudność konfiguracji.

Nasze zasoby były domyślnie dostępne pod adresem [http://*localhost*](http://localhost)jest to adres odnoszący się do naszego komputera

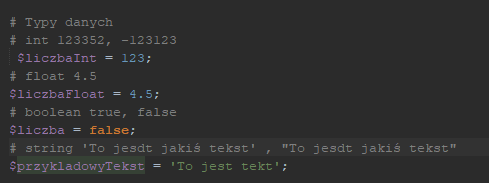
**Domyślne pliki**

W przypadku gdy nie podamy dokładnej ścieżki do pliku html lub php na serwerze np. <http://localhost/strona.html> serwer będzie szukał na serwerze domyślny plik zdefiniowany w konfiguracji serwera, można samemu skonfigurować jakiego pliku serwer ma szukać, domyślnie jest to plik index.html i index.php

**Typy danych w PHP**

Zaczęliśmy pisać skrypty php od definiowania zmiennych (zakładek w pamięci komputera na dane które chcemy używać)

Żeby zdefiniować zmienną w php używamy znaku dolara ($) po którym podajemy nazwę np.



Wielkości zmiennych nie są bardzo restrykcyjne w php gdyż serwer zajmuje tyle samo miejsca dla wartości logicznej co dla float. Dzieje się to z tego powodu, że php umożliwia dynamiczne zmienianie typu, każda zmienna może przyjmować dowolne stany od inta, przez float, boolean do ciągu znaków(string).

**Int**

**32-bit PHP:**

* Int może mieć wielkości od **-2,147,483,648** do **2,147,483,647** (~ ± 2 billion)

**64-bit PHP:**

* Int może mieć wielkość od **-9,223,372,036,854,775,808** do **9,223,372,036,854,775,807** (~ ± 9 quintillion)

**Float**

Prezyja Float jest do 1.11e-16 czyli 0.000000000000000111

**Boolean**

Wartości logiczne

**String**

Definiujemy w apostrofach lub w cudzysłowach

* Apostrofy nie interpretują specjalnych znaczników zmienna $liczba nie zostanie przetworzona



* Cudzysłowie umożliwiają wpisanie dodatkowych znaczników które zostaną zinterpretowane w ciągu znaków. Zawartość zmiennej $przykladowyTekst zostanie przekopiowana do ciągu znaków.



* Aby zapobiec w przetwarzania specjalnych znaczników jak dolar w cudzysłowie, musimy uprzedzić specjalny znacznik back-slashem. IDE wykryło, że przed zmienną $przykładowyText jest back-slash i potraktowało następujący po nim znak dolara jako zwykły tekst, nie jako odnośnik do zmiennej.



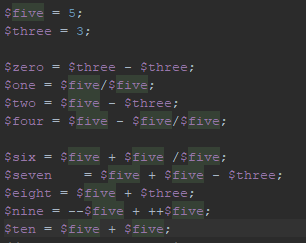
* Możemy używać obu apostrofów i cudzysłowów w jednym ciągu znaków. Jeśli zaczniemy deklarację ciągu znaków od apostrofa (‘), cudzysłowie które występują wewnątrz apostrofów będą interpretowane jako zwykły tekst gdyż php będzie oczekiwało końca ciągu znaków oznaczonego drugim apostrofem



**Operatory w PHP**

Wyróżniamy kilka głównych operatorów są to dodawanie(+), odejmowanie(-), mnożenie(\*), dzielenie(/), wyciąganie reszty z dzielenia całkowitego(%), inkrementacja(++), dekrementacja(--)

Aby to przećwiczyć wykonaliśmy zadanie: “Zakładając, że mamy tylko same piątki i same trójki, obliczmy wszystkie inne liczby od 0 do 10, używając tylko tych dwóch liczb i podanych wyżej operatorów.



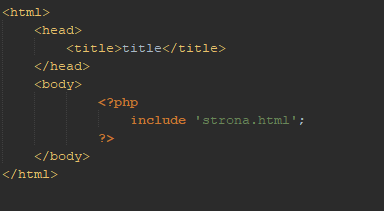
**Komunikacja PHP z przeglądarką**

Php jest wykonywany na serwerze i wszystkie instrukcję które się w nim zawierają są całkowicie tajne. Przeglądarka może nawet nic nie wiedzieć o tym, że na serwerze jest jakieś php, ona otrzymuje tylko zrozumiały i czytelny dla siebie kod HTML. PHP i HTML są przetwarzane w dwóch całkowicie innych przestrzeniach. Html w przeglądarce, php na serwerze. Żeby zakomunikować coś z php do przeglądarki używamy polecenia **echo** lub **var\_dump**. Podobnie jest z Bogiem, nie da się udowodnić, że on istnieje, ale wierzymy, że tak jest i Bóg się z nami komunikuje. Ludzie spisali wszystko co Bóg do nich mówił i stąd mamy Biblię. Później nawet przysłał Jezusa by on powiedział więcej.

**HTML pisany na przemiennie z kodem PHP**

Jeśli stworzymy plik z zakończeniem php, mamy możliwość pisania w nim kod php jak i kod html. Przymusem jest to, że wstawiając jakiś kod php pomiędzy kod html, musimy pamiętać o napisaniu tagu zakończenia skryptu php instrukcja **Include**

Zobaczyliśmy, że do pliku php, mamy możliwość do (**?>**), gdyż następujące później tagi html, serwer będzie próbował interpretować jako php, którym nie jest. Na przykładzie poniżej widać jak można dopisać jakiś skrypt php w strukturze html



**Include**

Mamy też możliwość by dynamicznie doładować treść innego pliku np. Pliku strona.html, do wewnątrz pliku php i umieścić jego treść dokładnie w tym miejscu gdzie zażądaliśmy załączenie strona.html (Stworzyliśmy skrypt php z instrukcją include wewnątrz tagu <body></body>)

**Bootstrap**

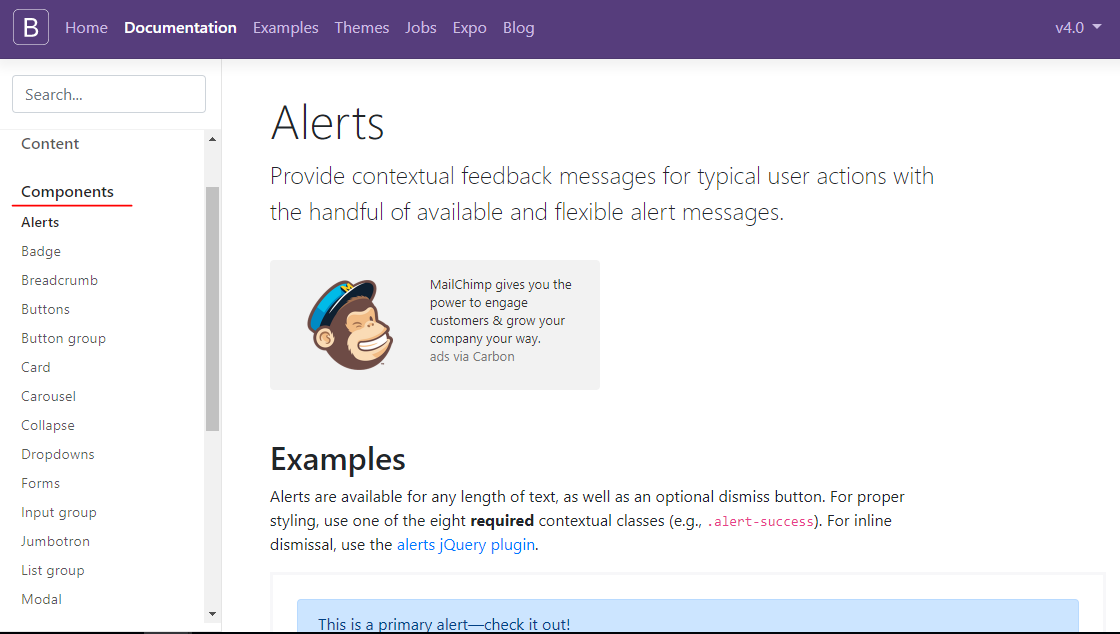
Spojrzeliśmy na zestaw narzędzi i efektów do generowanie wyglądu strony html. Użyliśmy szablonu bootstrap aby osadzić tam naszą stronę pizzerii.

Bootstrap dzięki temu, że już ma zdefiniowane najczęściej używane instrukcje css i js, znacznie przyspiesza pracę.

Mamy wiele różnych komponentów na stronę gotowych do użycia, zachęcam do zapoznania się z listą komponentów w dokumentacji bootstrapa. Zobaczcie obrazek poniżej

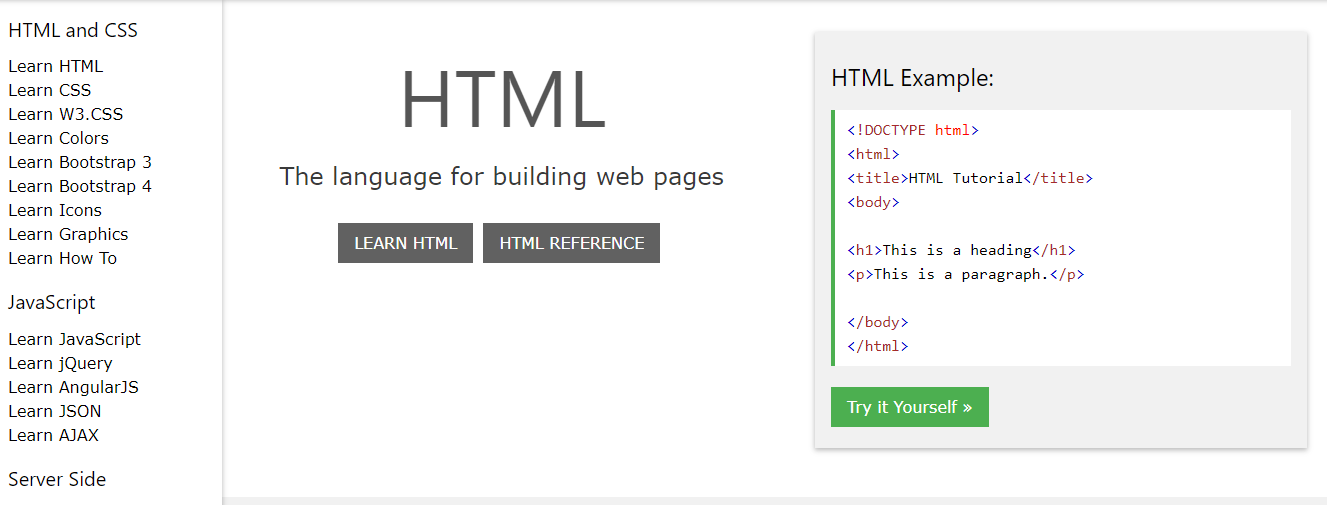
Link do strony bootstrapa:

<https://getbootstrap.com/docs/4.0/components/alerts/>



**W3SCHOOL**

Spojrzeliśmy również na inne źródła wiedzy na temat podstaw HTML, CSS, PHP na stronie <https://www.w3schools.com/>



Jest to bardzo interaktywna strona internetowa która definiuje co robi jakaś instrukcja HTML, CSS, PHP i daje możliwość by zobaczyć ją w działaniu, wystarczy kliknąć 

**Pasja Informatyki**

Na youtube jest kanał na temat programowania CSS, HTML możecie powtórzyć sobie ten temat właśnie tam

Materiały na temat css

<https://www.youtube.com/watch?v=RJEKMbD_kEk&list=PLOYHgt8dIdow6b2Qm3aTJbKT2BPo5iybv>

Materiały na temat html

<https://www.youtube.com/watch?v=1M0YXFW31hg&list=PLOYHgt8dIdox9Qq3X9iAdSVekS_5Vcp5r>

**Praca domowa:**

Chciałbym abyście zapoznali się z listą komponentów bootstrapa, nie musicie ich używać, ani uczyć na pamięć. Chciałbym tylko byście spojrzeli jak się co nazywa i mieli generalną wiedzę o istnieniu takiego efektu. Proszę poświęćcie na to tak z 30 minut. To naprawdę pomoże później gdy będziecie chcieli coś znaleźć, a już będziecie wiedzieć, że coś takiego kiedyś widzieliście.

Znajdziecie je na stronie <https://getbootstrap.com/docs/4.0/components/alerts/> lub klikając w link na komponencie poniżej.

Wymienione komponenty na stronie bootstrapa to:

* [Alerts](https://getbootstrap.com/docs/4.0/components/alerts/)
* [Badge](https://getbootstrap.com/docs/4.0/components/badge/)
* [Breadcrumb](https://getbootstrap.com/docs/4.0/components/breadcrumb/)
* [Buttons](https://getbootstrap.com/docs/4.0/components/buttons/)
* [Button group](https://getbootstrap.com/docs/4.0/components/button-group/)
* [Card](https://getbootstrap.com/docs/4.0/components/card/)
* [Carousel](https://getbootstrap.com/docs/4.0/components/carousel/)
* [Collapse](https://getbootstrap.com/docs/4.0/components/collapse/)
* [Dropdowns](https://getbootstrap.com/docs/4.0/components/dropdowns/)
* [Forms](https://getbootstrap.com/docs/4.0/components/forms/)
* [Input group](https://getbootstrap.com/docs/4.0/components/input-group/)
* [Jumbotron](https://getbootstrap.com/docs/4.0/components/jumbotron/)
* [List group](https://getbootstrap.com/docs/4.0/components/list-group/)
* [Modal](https://getbootstrap.com/docs/4.0/components/modal/)
* [Navs](https://getbootstrap.com/docs/4.0/components/navs/)
* [Navbar](https://getbootstrap.com/docs/4.0/components/navbar/)
* [Pagination](https://getbootstrap.com/docs/4.0/components/pagination/)
* [Popovers](https://getbootstrap.com/docs/4.0/components/popovers/)
* [Progress](https://getbootstrap.com/docs/4.0/components/progress/)
* [Scrollspy](https://getbootstrap.com/docs/4.0/components/scrollspy/)
* [Tooltips](https://getbootstrap.com/docs/4.0/components/tooltips/)