**HTML – podstawy**

**Losowe informacje:   
„!”** to skrót do zrobienia boilerplate  
**„lorem”** to skrót do utworzenia zdania lorem

**Podstawowe elementy:**

**<!DOCTYPE html>** - formułka na początku każdego pliku   
**<html> </html>** - wszystko oprócz powyższego ma się w tym znajdować   
**<head> </head>** - rzeczy niewidoczne. Można zapisać css i tytuł strony  
**<title> </title>** - tytuł strony wyświetlany w googlu, albo w kartach   
**<body>** - wszystko zasadniczo co jest wypisane na stronie

**<h1>,<h2>…<h3>** - tytuły. Im mniejsza liczba tym mniej ważny podtytuł   
**<p>** - nowy paragraf

**<b>** - pogrubienie, „bald”

**<i>** - kursywa

**-listy-**

**<ul>** - nieuporządkowana lista   
**<ol>** - uporządkowana lista  
**<li>** - element listy

**(shitft+alt+down/up arrow)** – pozwala szybko kopiować elementy listy

**Random** – można tworzyć listy w listach (<ol> <li> <ol> <li> <///>)

**-linki-**

**<a>** - anchortag. Opcja dodania linku.   
<a href=”{link}”></a>  
Jeśli link jest do strony www potrzeba dodać http://www przed nim

Jeśli link jest do pliku w tej samej przestrzeni dodajemy w href nazwę i rozszerzenie pliku

Link może przechodzić do pod folderu na zasadzie „{foler}/{nazwa pliku i rozszerzenie}”

**-zdjęcia-**

**<img src =”{lokacja zdjęcia}” alt=”photo descritption in text”>**Teoretycznie można zmieniać w html’u wielkość zdjęcia (przez atrybut width), ale to zła praktyka bo lepiej na css

**-komentarze-   
<!-- .. {tekst} .. -->** - {tekst} zawiera komentarz niewidzialny dla przeglądarki   
„command + /” – skrót do wykomentowania linii

**HTML – rozszerzone**

**-inlne/block-**

**Inline elements** – mogą się mieścić w jednej linii. Nie wypychają siebie nawzajem   
**Block elements** – każdy z nich zajmuje samemu pełne miejsce (np. pełną linię nawet jak jest pusta)

**<div>** - pozwala na grupowanie elementów. Wymuszone dodanie atrybutu „block elements” (zajmuje cały blok)  
**<span>** - pozwala na oddzielne dostosowywanie elementów oznaczonego tym elementem. Typ generyczny zasadniczo

**-mniej używane elementy-**

**<hr>** - linia oddzielająca dwie części tekstu od siebie. Nie ma klamry zamykającej

**<br>** - wymusza nową linię. Używane wewnątrz <p> </p>. Nie ma klamry zamykającej

**<sup>** - element z mniejszym fontem NAD wyrażeniem  
**<sub>** - element z mniejszym fontem POD wyrażeniem

**-entity codes-**<https://html.spec.whatwg.org/multipage/named-characters.html> – wszystkie wyróżnione znaki i ich kody   
**&{kod};** - poprawna struktura odwołania się do elementu

**-semantyczność-**

Zamiast korzystać z armii <div>’ów możemy użyć zamiennie elementów które w nazwie mają swoje działanie **<main>** - główna zawartość strony. Niepowtarzalna względem innych stron

**<header>** - taka trochę introdukcja. Może być używane wielokrotnie w osobnych artykułach  
**<nav>** - reprezentacja linków

**<footer>** - „dolna” część strony. Może być używane wielokrotnie w osobnych artykułach

**<section>** - oznacza sekcję

**<article>** - taka „nadsekcja”

**<time>** - oznacza czas i datę 

**-emmet-   
main>section>h1\*5 –** przykład struktury która utworzy gotową formę   
[**https://docs.emmet.io/cheat-sheet/**](https://docs.emmet.io/cheat-sheet/)- cheat sheet

**>** - tworzy dzieci elementów nadrzędnych   
**+** - tworzy elementy równe sobie   
**\*** - tworzy kilka elementów tego samego typu

**$** - tworzy automatyczną numerację   
**nav>ul>li\*5>a[href=www.site$.com]{Click me}**

**HTML – jeszcze bardziej rozszerzone**

**-tablice-**

Używane po to, żeby wyświetlić i porównać ze sobą przedmioty względem wybranych jednostek

**<table>** - tworzy i ogranicza tabelę

**<td>** - oznacza pojedynczą komórkę - data

**<tr>** - oznacza pojedynczy wiersz - row  
**<th>** - oznacza komórkę z nagłówkiem. Musi zapełniać pierwszy wiersz - header

**Tworzenie sekcji tablicy** – używa się, aby rozdzielić jedne części od drugich w tablicy. Niczym sekcje w stronie   
**<thead>** - oznaczenie nagłówków   
**<tbody>** - oznaczenie wnętrza tabeli  
**<tfoot>** - oznaczenie ostatniego wiersza

**rowspan** – atrybut <th> określający jego wysokość.   
**colspan** – atrybut <th> określający na ile się rozbija części   
Jeśli nasz drugi <tr> zawiera w sobie <th> dodając atrybut „colspan” w pierwszych <th> utworzymy pod kolumny

**-formy-**

**<form action=”{adres}” method=”{rodzaj http request}”>** - kontener gdzie umieszczamy formy html-owskie. Precyzuje gdzie formy powinny wysłać dane

**<input type=”{typ}” id = „{}” name=”{używane do identyfikacji pochodzenia przesyłanych elementów}” placeholder=”{napis wyświetlany przed wpisaniem własnego tekstu}”>** - używane do przekazywania informacji. Rodzaje to m.in.:

**text** – zwykły text

**password** – hasło. Zasłonięte literki  
**number** – możliwość wpisywania tylko cyfer. Max i min występuje

**checkbox** – element do zaznaczania (ptaszkiem)

**<label for=”{wpisujemy id inputu}”>** - Co ma być wypisane, ale tym samym połączone z odpowiednią formą

**-buttons-**

**<button type=”{submit/button}”>** - Tworzy przycisk. Pomiędzy klamerkami można zapisać co ma być w nim zapisane. Bazowo ustawiony na sumbit. Rodzaje:  
**submit** – wysyła formę poprzez podane atrybuty   
**button** – można klikać, ale bazowo nic nie robi

**<input type=”submit” value=”{whatever we wanna show inside of button}”>** - mniej optymalny sposób zrobienia przycisku

**More inputs:**

**<input type=”checkbox” name=”” id=”’ checked/or not write it>** - element do zaznaczania ptaszkiem. Jeśli zaznaczony to uwzględniany w submicie, jak nie to nie

**<input type=”radio” name=”{to samo dla całej grupy}” id=”’ value=”{jaka wartość wysyłana}”** – element do zaznaczania pojedynczych wyników

**Select** – wymaga w sobie **options**. Select działa tak samo jak i input, ale ma też zakończenie. Pomiędzy nimi należy umieścić pola do wybrania z rosuwanego menu czyli**:**

**<option value=”{to co wysyłane}” selected/or not write it></>** - oznacza pojedyncze elementy do wyświetlenia. Czasem tworzony jest dummy element wyświetlany jako bazowy na początku

**<input type=”range” max=”” min=”” value=”{pozycja startowa}”>** - suwak

**-Validations-**

**<input … required>** - warunek, który sprawia że pole jest wymagane

**<input … maxlength=”” minlength=””>** - używane do sprawdzania długości czy jest ona oczekiwanej

Typ email I url ma wbudowane sprawdzanie paternu