

Temat: Zapoznanie z narzędziami wykorzystywanymi do tworzenia zaawansowanych aplikacji webowych.

1. **Visual Studio Code** to edytor kodu zoptymalizowany pod kątem tworzenia i debugowania nowoczesnych aplikacji internetowych.
2. **Google Chrome** jest obecnie najbardziej rozpowszechnioną przeglądarką internetową. Posiada ona narzędzia developerskie przydatne w procesie tworzenia aplikacji internetowych.
3. **Node.js** jest wieloplatformowym środowiskiem uruchomieniowym o otwartym kodzie do tworzenia aplikacji typu server-side napisanych w języku JavaScript. Przyczynił się do stworzenia paradygmatu „JavaScript everywhere”, umożliwiając programistom tworzenie aplikacji w obrębie jednego języka programowania zarówno po stronie przeglądarki, jak i serwera.

Temat: Frameworki i biblioteki - różnice i podobieństwa.

1. Framework:

- stanowi szkielet budowy aplikacji
- definiuje on działanie każdego elementu
- w jego skład wchodzi biblioteki i komponenty
- nie zawsze korzystamy z wszystkich jego elementów
- nie zawsze jest wydajny

2. Biblioteka:

- wykonuje określone działanie
- jest bazą / podstawą rozbudowywaną przez programistę
- często jest samodzielnym elementem lub elementem grupy / frameworka
- korzystamy tylko z tych elementów, które są nam potrzebne do danego projektu

Temat: Wprowadzenie do frameworka Bootstrap.

1. Framework Bootstrap:

- jeden z najpopularniejszych gotowych szkieletów strony internetowej
- ułatwia tworzenie wizualnej części strony internetowej
- umożliwia szybkie budowanie responsywnych stron
- jest darmowy, licencja pozwalająca na jego używanie w komercyjnych projektach
- daje możliwość stylizowania z zastosowaniem gotowych klas
- obsługuje wtyczki (podobnie jak jQuery)
- stworzony przez programistów Twittera Marka Otto i Jacoba Thorntona
- udostępniony w 2011 r.
- posiada wiele przydatnych poradników

2. **Responsywne projektowanie stron internetowych (ang. responsive web design, RWD)** – technika projektowania strony WWW, aby jej układ dostosowywał się samoczynnie do rozmiaru okna przeglądarki, na której jest wyświetlany. Strona utworzona tą techniką jest uniwersalna: wyświetla się dobrze na dużych ekranach oraz na smartfonach i tabletach: zmienia swój rozmiar oraz układ, obsługuje różne rozdzielczości w zależności od przekątnej ekranu, na jakim strona internetowa ma być wyświetlona.

Temat: Siatka Bootstrapa.

1. Siatka Bootstrapa:

- składa się maksymalnie z 12 kolumn rozdzielonych odstępami
- dzieli ekran na dwanaście równych części niezależnie od szerokości użytego ekranu
- pozwala tworzyć kolumny, których szerokość może wahać się od 1/12 do 12/12 ekranu

2. Bootstrap 5 definiuje 6 szerokości ekranu, przy których może nastąpić zmiana układu siatki:

- extra small, do 575 px – klasa ".col-"
- small, od 576 px do 767 px – klasa ".col-sm-"
- medium, od 768 px do 991 px – klasa ".col-md-"
- large, od 992 px do 1199 px – klasa ".col-lg-"
- extra large, od 1200 px do 1399 px – klasa ".col-xl-"
- extra extra large – od 1400 px – klasa ".col-xxl-"

Temat: Obsługa formularzy w frameworku Bootstrap.

1. Framework Bootstrap wspiera tworzenie formularzy poprzez:

- poprawienie wyglądu w porównaniu do domyślnych ustawień przeglądarek internetowych
- ujednolicenie wyglądu pomiędzy różnymi przeglądarkami
- wprowadzenie trzech rozmiarów: small, default i large
- pływające etykiety
- mechanizm walidacji

Temat: Komponenty w frameworku Bootstrap.

1. Framework Bootstrap 5 dostarcza nam wiele gotowych komponentów. Najważniejsze z nich to:

- Navbar - menu główne strony
- Offcanvas - rozwijane menu boczne
- Breadcrumb - "okruszki chleba", nawigacja pomocnicza w kategoriach i podkategoriach
- Carousel - slider zdjęć i/lub tekstów
- Navs and tabs - zakładki wraz z menu pozwalającym przełączać się pomiędzy nimi
- Modal - okno modalne, np. do wyświetlenia informacji o cookies lub formularza newslettera
- Accordion - mechanizm przydatny przy tworzeniu sekcji najczęściej zadawanych pytań
- Alerts - powiadomienia służące do przekazania komunikatu o powodzeniu, ostrzeżeniu, błędzie lub informacji pomocniczej
- Pagination - paginacja, podział na podstrony

Temat: Wprowadzenie do biblioteki jQuery.

1. jQuery:

- biblioteka JavaScriptu
- udostępniona w 2006 roku
- powstała w celu opisanym w logo: write less, do more
- pozwala tworzyć nowoczesne i funkcjonalne strony
- posiada bardziej intuicyjny wybór elementów niż w JS
- umożliwia "chwytnie" elementów za pomocą selektorów w stylu CSS
- w łatwiejszy sposób pozwala na podmianę, ukrycie, usunięcie oraz nadanie klasy do elementu
- pozwala na uaktualnianie strony o nowe efekty bez przebudowy jej kodu
- automatycznie dba o obsługę funkcjonalności i sprawia, że będą działać nawet w starszych przeglądarkach, dzięki czemu programista może skupić się na mechanice strony zamiast obsługi zdarzeń

2. Podłączenie jQuery do strony:

- przez pobranie wersji skompresowanej lub nieskompresowanej
- lokalnie lub poprzez serwer CDN
- przez podanie ścieżki do pliku, która jest umieszczona pomiędzy znacznikami `<script></script>` w sekcji head lub przed zamknięciem body

Temat: Wyszukiwanie elementów w jQuery.

1. jQuery umożliwia "chwytnie" elementów za pomocą selektorów w stylu CSS.
2. Z jQuery można korzystać, używając pełnej nazwy funkcji, `jQuery()`, jak również formy skróconej, `$()`.
3. Przykłady "chwytnia" elementów HTML za pomocą jQuery:
 - `$('li')` - elementy listy
 - `$('#menu')` - element o id "menu"
 - `$('.active')` - elementy posiadające klasę "active"
 - `$('[href="#"]')` - elementy posiadające atrybut href="#"
4. Metody pomocne w chwytniu wybranych elementów strony:
 - `.first()` - pierwszy element z wybranych
 - `.last()` - ostatni element z wybranych
 - `.even()` - nieparzyste elementy zbioru
 - `.odd()` - parzyste elementy zbioru
 - `.eq(nr)` - elementy o podanym numerze indeksu
 - `.lt(nr)` - elementy o numerze indeksu mniejszym niż podany
 - `.gt(nr)` - elementy o numerze indeksu większym niż podany
 - `.not(selektor)` - wszystkie elementy z wyjątkiem wskazanego
 - `.find(selektor)` - wyszukuje wewnątrz danego elementu inny element określony przez selektor
 - `.filter(fn*)` - filtruje listę elementów przez zadane kryterium
5. Przy użyciu np. selektora `.eq()` zostaje pobrana cała kolekcja elementów, a to, który zostanie zastosowany, określa zdefiniowany indeks. Użycie np. kodu `$('button').eq(1)` sprawi, że z kolekcji wszystkich przycisków wybrany zostanie drugi.

Temat: Pobieranie zawartości elementów w jQuery.

1. Metoda `.html()`:
 - pobiera lub ustawia kod HTML elementu
 - przykłady:

```
const $lista_ul = $('ul').html();  
const $lista_li = $('li').html();
```

2. Metoda `.text()`:

- pobiera lub ustawia tekst elementu
- przykład:

```
let $lista_ul = $('ul').text();  
let $lista_li = $('li').text();
```

3. Połączenie metod `.html()` oraz `.text()`:

- przykład:

```
$(function () {  
  let $lista_ul = $('ul').html();  
  $('ol').append($lista_ul);  
  
  $lista_ul = $('ul').text();  
  $('p').append($lista_ul);  
});
```

Temat: Wstawianie elementów w jQuery.

1. Metody wykorzystywane do wstawiania elementów w jQuery:

- `.append()` - wstawia treść na końcu każdego elementu w zestawie dopasowanych elementów
- `.appendTo()` - wstawia każdy element w zestawie dopasowanych elementów na koniec elementu docelowego
- `.prepend()` - wstawia treść na początek każdego elementu w zestawie dopasowanych elementów
- `.prependTo()` - wstawia każdy element w zestawie dopasowanych elementów na początek elementu docelowego
- `.after()` - wstawia treść po każdym elemencie w zestawie pasujących elementów
- `.before()` - wstawia treść przed każdym elementem w zestawie dopasowanych elementów
- `.insertAfter()` - wstawia każdy element w zestawie dopasowanych elementów po elemencie docelowym
- `.insertBefore()` - wstawia każdy element w zestawie dopasowanych elementów przed elementem docelowym

Temat: Modyfikowanie i usuwanie elementów w jQuery.

1. Metody wykorzystywane do modyfikowania i usuwania elementów elementów w jQuery:

- `.attr()` - pobiera wartość atrybutu dla pierwszego elementu w zestawie dopasowanych elementów lub ustawia jeden lub więcej atrybutów dla każdego dopasowanego elementu
- `.removeAttr()` - usuwa atrybut z każdego elementu w zestawie dopasowanych elementów
- `.prop()` - pobiera wartość właściwości pierwszego elementu w zestawie dopasowanych elementów lub ustaw jedną lub więcej właściwości dla każdego dopasowanego elementu
- `.removeProp()` - usuwa właściwość z zestawu dopasowanych elementów
- `.css()` - pobiera wartość właściwości stylu dla pierwszego elementu w zestawie dopasowanych elementów lub ustawia jedną lub więcej właściwości CSS dla każdego dopasowanego elementu
- `.addClass()` - dodaje określone klasy do każdego elementu w zestawie dopasowanych elementów
- `.hasClass()` - sprawdza, czy któryś z dopasowanych elementów ma przypisaną daną klasę
- `.removeClass()` - usuwa jedną lub wiele klas lub wszystkie klasy z każdego elementu w zestawie dopasowanych elementów
- `.toggleClass()` - dodaje lub usuwa jedną lub wiele klas z każdego elementu w zestawie dopasowanych elementów, w zależności od obecności klasy lub wartości argumentu stanu
- `.replaceWith()` - zastępuje każdy element w zestawie dopasowanych elementów dostarczoną nową treścią i zwraca zestaw elementów, który został usunięty
- `.replaceAll()` - zastępuje każdy element docelowy zestawem dopasowanych elementów
- `.remove()` - usuwa zestaw dopasowanych elementów z modelu DOM
- `.empty()` - wsuwa wewnętrzny kod HTML wszystkich dopasowanych elementów z modelu DOM
- `.each()` - pobiera wiele elementów i na każdym z nich po kolei wykonuje wybrane operacje

Temat: Obsługa zdarzeń w jQuery.

1. W jQuery możemy łatwo podpiąć procedurę obsługi zdarzeń do wybranych elementów. Kiedy następuje określone zdarzenie, zostaje wykonana przypisana do niego funkcja.
2. Aby nasłuchiwać zdarzenie do pobranego lub utworzonego elementu, używamy metody będącej nazwą zdarzenia lub używamy metody .on(), do której przekazujemy procedurę obsługi zdarzeń.

Przykład:

```
$('p').click( function() {  
    $(this).remove();  
});  
  
$('p').on('click', function() {  
    $(this).remove();  
});
```

3. Zdarzenia w jQuery możemy podzielić na związane z:

- elementami interfejsu
- klawiaturą
- myszą
- użyciem formularza
- przeglądarką

Temat: Zdarzenia elementów interfejsu i przeglądarki w jQuery.

1. Zdarzenia elementów interfejsu w jQuery:

- `.focus()` - wiąże obsługę zdarzenia focus lub wywołuje to zdarzenie na elemencie; zdarzenie występuje, gdy wybierzemy element za pomocą myszki lub klawisza TAB
- `.change()` - wiąże obsługę zdarzenia change lub wywołuje to zdarzenie na elemencie; zdarzenie występuje, gdy edytujemy element
- `.blur()` - wiąże obsługę zdarzenia blur lub wywołuje to zdarzenie na elemencie; zdarzenie występuje, gdy opuszczamy aktywny element

2. Zdarzenia przeglądarki w jQuery:

- `.scroll()` - wiąże obsługę zdarzenia scroll lub wywołuje to zdarzenie na elemencie; zdarzenie występuje, gdy użytkownik przewija zawartość okna przeglądarki
- `.resize()` - wiąże obsługę zdarzenia resize lub wywołuje to zdarzenie na elemencie; zdarzenie występuje, gdy użytkownik zmienia rozmiar okna przeglądarki

Temat: Zdarzenia klawiatury i myszy w jQuery.

1. Zdarzenia klawiatury w jQuery:

- `.keydown()` - wiąże obsługę zdarzenia `keydown` lub wywołuje to zdarzenie na elemencie; zdarzenie występuje, gdy naciśniemy klawisz na klawiaturze klawisze niedrukowalne, jak Shift, Esc, Delete
- `.keypress()` - wiąże obsługę zdarzenia `keypress` lub wywołuje to zdarzenie na elemencie; zdarzenie występuje, gdy naciśniemy klawisz na klawiaturze klawisze drukowalne, jak litery, cyfry itp.
- `.keyup()` - wiąże obsługę zdarzenia `keyup` lub wywołuje to zdarzenie na elemencie; zdarzenie występuje, gdy zwolnimy klawisz na klawiaturze

2. Zdarzenia myszy w jQuery:

- `.click()` - wiąże obsługę zdarzenia `click` lub wywołuje to zdarzenie na elemencie; zdarzenie występuje, gdy klikniemy przycisk myszki
- `.mouseover()` - wiąże obsługę zdarzenia `mouseover` lub wywołuje to zdarzenie na elemencie; zdarzenie występuje, gdy umieścimy wskaźnik myszy w obrębie wybranego elementu
- `.mouseout()` - wiąże obsługę zdarzenia `mouseout` lub wywołuje to zdarzenie na elemencie; zdarzenie występuje, gdy usuniemy wskaźnik myszy z wybranego elementu

Temat: Obiekt event w jQuery.

1. Obiekt event w jQuery dostarcza informacji o zaistniałym zdarzeniu. Użyta w wywołaniu funkcja pozwala odwołać się do obiektu zdarzenia.

2. Właściwości obiektu event:

- `.currentTarget` - bieżący element DOM w fazie propagacji zdarzenia
- `.data` - opcjonalny obiekt danych przekazywany do metody zdarzenia, gdy jest powiązany bieżący skrypt obsługi zdarzenia
- `.delegateTarget` - element, do którego dołączono aktualnie wywoływaną procedurę obsługi zdarzeń jQuery
- `.metaKey` - wskazuje, czy podczas uruchomienia zdarzenia naciśnięto klawisz Meta; klawisz Meta to Windows na klawiaturze Windows oraz Command na klawiaturach Macintosh
- `.namespace` - przestrzeń nazw określona, gdy zdarzenie zostało wyzwolone
- `.pageX` / `.pageY` - pozycja myszy względem lewej/górnej krawędzi dokumentu
- `.relatedTarget` - drugi element DOM biorący udział w zdarzeniu, jeśli istnieje
- `.result` - ostatnia wartość zwrócona przez procedurę obsługi zdarzenia, która została wywołana przez to zdarzenie, chyba że wartość była niezdefiniowana.
- `.target` - element DOM, który zainicjował zdarzenie
- `.timestamp` - różnica w milisekundach między czasem utworzenia zdarzenia przez przeglądarkę a 1 stycznia 1970 r.
- `.type` - opisuje typ zdarzenia, np. `keyup`
- `.which` - wskazuje klawisz lub przycisk, który został naciśnięty

3. Metody obiektu event:

- `.isDefaultPrevented()` - zwraca informację, czy funkcja `event.preventDefault()` została kiedykolwiek wywołana na tym obiekcie zdarzenia
- `.isImmediatePropagationStopped()` - zwraca informację, czy funkcja `event.stopImmediatePropagation()` została kiedykolwiek wywołana dla tego obiektu zdarzenia
- `.isPropagationStopped()` - zwraca informację, czy funkcja `event.stopPropagation()` została kiedykolwiek wywołana dla tego obiektu zdarzenia
- `.preventDefault()` - jeżeli ta metoda zostanie wywołana, domyślna akcja zdarzenia nie zostanie wywołana
- `.stopImmediatePropagation()` - uniemożliwia wykonanie pozostałych procedur obsługi i zapobiega rozprzestrzenianiu się zdarzenia w drzewie DOM
- `.stopPropagation()` - zapobiega rozprzestrzenianiu się zdarzenia w drzewie DOM, uniemożliwiając powiadomienie jakichkolwiek nadrzędnych procedur obsługi o zdarzeniu

Temat: Efekty w jQuery.

1. Efekty podstawowe:

- `.show()` - wyświetla dopasowane elementy
- `.hide()` - ukrywa dopasowane elementy
- `.toggle()` - wyświetla lub ukrywa dopasowane elementy

2. Efekty przeźroczystości:

- `.fadeIn()` - zmienia widoczność dopasowanych elementów od przeźroczystości do pełnej widoczności
- `.fadeOut()` - zmienia widoczność dopasowanych elementów od pełnej widoczności do przeźroczystości
- `.fadeTo()` - ustawia przezroczystość dopasowanych elementów
- `.fadeToggle()` - wyświetla lub ukrywa dopasowane elementy, animując ich przeźroczystość

3. Efekty slajdów:

- `.slideDown()` - wyświetla dopasowane elementy ruchem przesuwным
- `.slideUp()` - ukrywa dopasowane elementy ruchem przesuwным
- `.slideToggle()` - wyświetla lub ukrywa dopasowane elementy, animując ich ruch przesuwny

Temat: Animacje w jQuery.

1. Do tworzenia niestandardowych animacji zestawu właściwości CSS w jQuery używamy metody `.animate()`.
2. W jQuery animowane mogą być wartości stylu wyrażane w postaci liczby, np. `width`, `padding`, `font-size`.

3. Właściwości metody `.animate(properties [, duration ] [, easing ] [, complete ])`:

- `properties` - właściwości CSS (jedna lub kilka)
- `duration` - czas trwania w milisekundach lub za pomocą słów "fast" lub "slow"
- `easing` - sposób przebiegu animacji w czasie: swing (zwiększenie prędkości w środku i zmniejszenie na początku i na końcu) lub linear (stała prędkość od początku do końca)
- `complete` - funkcja wywołania zwrotnego, gdy animacja się zakończy

4. W przypadku właściwości CSS, których nazwy składają się z wielu słów oddzielonych od siebie znakiem łącznika, np. `border-radius`, w połączeniu z metodą `.animate()` używamy notacji camelCase - pierwsze słowo rozpoczynamy od małej litery, a każde następne od dużej, np. `borderRadius`.

5. Jeżeli jednocześnie są animowane dwie lub więcej właściwości CSS, muszą one znajdować się w nawiasie klamrowym i być oddzielone od siebie znakiem przecinka.

6. Użycie zapisu `+=` doda odpowiednią wielkość do aktualnej, a `-=` odejmie.

Temat: Obsługa formularzy w jQuery.

1. Najczęściej wykorzystywane selektory:

- `:button` - elementy `<button>` i `<input>` z atrybutem `type`, który ma wartość `button`
- `:checkbox` - element `<input>`, którego atrybut `type` ma wartość `checkbox`
- `:disabled/:enabled` - elementy, które zostały wyłączone/włączone
- `:input` - elementy `<button>`, `<input>`, `<select>` oraz `<textarea>`
- `:password` - element `<input>`, którego atrybut `type` ma wartość `password`
- `:radio` - element `<input>`, którego atrybut `type` ma wartość `radio`
- `:submit` - element `<input>` bądź `<button>`, którego atrybut `type` ma wartość `submit`
- `:text` - element `<input>`, którego atrybut `type` ma wartość `text`

2. Metody wykorzystywane w pracy z formularzami:

- `.val()` - pobiera bieżącą wartość pierwszego elementu w zestawie dopasowanych elementów lub ustawia wartość każdego dopasowanego elementu
- `.is()` - sprawdza bieżący dopasowany zestaw elementów względem selektora, elementu lub obiektu jQuery i zwraca wartość `true`, jeśli przynajmniej jeden z tych elementów pasuje do podanych argumentów

Temat: Obsługa metod AJAX w jQuery.

1. **AJAX - Asynchroniczny JavaScript i XML (ang. Asynchronous JavaScript and XML)** – technika tworzenia aplikacji internetowych, w których interakcja użytkownika z serwerem odbywa się bez przeladowywania całego dokumentu, w sposób asynchroniczny.
2. **JSON - (ang. JavaScript Object Notation):**
 - lekki tekstowy format wymiany danych komputerowych
 - bazuje na podzbiorze języka JavaScript
 - jest niezależny od konkretnego języka
3. **API (ang. application programming interface)** - zbiór reguł ściśle opisujący, w jaki sposób programy lub podprogramy komunikują się ze sobą.
4. Najczęściej wykorzystywane metody AJAX w jQuery:
 - `.ajax()` - wykonuje asynchroniczne żądanie HTTP (AJAX)
 - `.get()` - łąduje dane z serwera za pomocą żądania HTTP GET
 - `.post()` - wysyła dane do serwera za pomocą żądania HTTP POST
5. Właściwości metody `.ajax(url [, settings ])`:
 - url - adres URL, na który wysyłane jest żądanie
 - settings - zestaw par klucz/wartość, które konfigurują żądanie AJAX; wszystkie ustawienia są opcjonalne
6. Właściwości metod `.get(url [, data ] [, success ] [, dataType ])` i `.post(url [, data ] [, success ] [, dataType ])`:
 - url - adres URL, na który wysyłane jest żądanie
 - data - obiekt lub ciąg znaków wysyłany do serwera wraz z żądaniem
 - success - funkcja wywołania zwrotnego, która jest wykonywana, jeśli żądanie się powiedzie
 - dataType - typ danych oczekiwanych z serwera