

## Spis treści

1. POBIERANIE DANYCH .....	2
2. FUNKCJE KTÓRĘ ODPOWIADAJĄ ZA PRAWIDŁOWE WYŚWIETLANIE DANYCH W HTML.....	3
3. WYŚWIETLANIE DANYCH W HTML .....	4
4. LOGIKA STRONY W JAVA SCRIPT, WAŻNE!!!! .....	5
.....	7
.....	8
5. HOSTING I KWESTIE TECHNICZNE STRONY .....	8

# JAKUB ACHELIK = PLAN LEKCJI JAVASCRIPT DOKUMENTACJA

Na zdalnych napisałem plan lekcji w czystym js (myślałem nad php+mysql, ale GitHub Pages nie wspiera), który na podstawie godziny i dnia tygodnia wyświetla info o aktualnej lekcji, plan lekcji na cały dzień i ile minut zostało do zakończenia obecnej lekcji. Dane są pobierane z tablic w pliku js.

Dodanie obsługi przerw

**Opis** = Dodałem obsługę = "OBLICZANIE ILE ZOSTAŁO MINUT PRZERWY"

**Opis tech.** = dzięki instrukcji warunkowej na podstawie godziny i minuty (zmienna hour & minute) nadpisuje wartość zmiennej **minute**, a powstałą wartość przypisuje do nowej zmiennej **przerwa** i wyświetla się w miejscu zmiennej **roznica**.

***kod:***

```
//PRZERWA
if (hour == 8 && minute >= 45 && minute <= 49 == true) { //przerwa1
  minute = minute - 50;
  minute = -minute;
  var przerwa = minute;
  roznica = przerwa + ' min. do końca przerwy';
} else if (hour == 9 && minute >= 35 && minute <= 39 == true) { //przerwa2
  minute = minute - 40;
  minute = -minute;
  var przerwa = minute;
  roznica = przerwa + ' min. do końca przerwy';
} else if (hour == 10 && minute >= 25 && minute <= 39 == true) { //przerwa3
  minute = minute - 40;
  minute = -minute;
  var przerwa = minute;
  roznica = przerwa + ' min. do końca przerwy';
}
```

***efekt:***

### Aktualna lekcja:

Lekcja to: Działalność gospodarcza w branży informatycznej

<https://zsoitmiastko.webex.com/meet/eugeniuszlisiecki>

Nauczyciel: Eugeniusz Lisiecki

Dzień: Poniedziałek

Lekcja 4: 10:40 - 11:25

Aktualna godzina: 10:25

Zostało: 15 min. do końca przerwy

Następna lekcja: Język polski

## 1. POBIERANIE DANYCH

Opis : Sposób pobierania danych na stronę z pliku [tablice.js](#)

Opis Tech: W pliku tablice.js znajdują się 3 tablice :

- **lekcja** – tablica z nazwą przedmiotów
- **link** – tablica z linkami do zajęć zdalnych
- **nauczyciel** – tablica z Imieniem nazwiska i nauczyciela

Dane te są wyciągane z kodu za pomocą zmiennej np.

lekcja[0] to z tablicy "Administracja bazami danych"

```

const lekcja = [
  "Administracja bazami danych ", //0
  "Administracja sieciowymi systemami operacyjnymi ", //1
  "Działalność gospodarcza w branży infomatycznej ", //2
  "Język angielski zawodowy w branży informatycznej ", //3
  "Język angielski ", //4
  "Język niemiecki ", //5
  "Język polski ", //6
  "Matematyka ", //7
  "Programowanie aplikacji internetowych ", //8
  'Projektowanie i montaż lokalnych sieci komputerowych', //9
  'Sieci komputerowe', //10
  'Systemy baz danych', //11
  'Witryny i aplikacje internetowe', //12
  'Wychowanie fizyczne', //13
  'Historia i Społeczeństwo', //14
  'Lekcja wolna', //15
  '-----', //16
  'Wychowawcza', //17
  'Wolne' //18
];

> const link = [ ...
];

> const nauczyciel = [ ...
];

```

## 2.FUNKCJE KTÓRĘ ODPOWIADAJĄ ZA PRAWIDŁOWE WYŚWIETLANIE DANYCH W HTML

Opis: Zbiór funkcji strzałkowych, w pliku przedmioty.js, które definiują dane, które mają wyświetlać się w HTML:

-link do lekcji , nazwa lekcji, nauczyciel

```
//PRZEDMIOTY

const ABD = () => {
  document.getElementById("link").href = link[0];
  document.getElementById("link").innerHTML = link[0];
  document.getElementById("lekcja").innerHTML = 'Lekcja to: ' + lekcja[0];
  document.getElementById("nauczyciel").innerHTML = " Nauczyciel: " + nauczyciel[0];
};

const ASSO = () => {
  document.getElementById("link").href = link[0];
  document.getElementById("link").innerHTML = link[0];
  document.getElementById("lekcja").innerHTML = 'Lekcja to: ' + lekcja[1];
  document.getElementById("nauczyciel").innerHTML = " Nauczyciel: " + nauczyciel[0];
}
```

W następnej części kodu funkcja jest wywoływana za pomocą polecenia np. ASSO();

### 3.WYŚWIETLANIE DANYCH W HTML

Opis: Kod HTML nie jest skomplikowany i sprowadza się do przechwycenia danych z java script za pomocą ID (klas) (document.getElementById.innerHTML=zmienna). Ostylowanie w CSS nie wymaga większej uwagi, użyłem podstawowych selektorów, natomiast rozmieszczenie elementów zdefiniowane jest za pomocą właściwości FLEXBOX.

## HTML

```
<body>

  <div class="info">
    <h3>Aktualna lekcja: </h3>
    <p id="lekcja"></p>
    <a id="link" target="_blank" href=""></a>
    <p id="nauczyciel"></p>
    <p id="weekday"></p>
    <p id="godzina"></p>
    <p id="hour"></p>
    <p id="roznica"></p>

    <p id="next"></p>
  </div>

  <main>
    <p id="dzien">

    </p>

    <p id="godzina2"></p>
  >
  <div class="item item1">...
</div>

  <div class="item item2">...
</div>

</main>
```

```
<script src="../../godziny.js"></script>
<script src="../../przedmioty.js"></script>
<script src="../../tablice.js"></script>
<script src="../../main.js"></script>
<script src="index.js"></script>
</body>
```

## CSS

```
.info {
  text-align: left;
  border: 1px solid black;
  padding: 10px;
  width: 300px;
  margin-right: 20px;
  margin-bottom: 10px;
  background-image: url('img/white-logo-zsoit.png');
}

main {
  border: 1px solid black;
  background-image: url('img/white-background-logo-zsoit.png');
  padding: 10px;
  width: 300px;
  margin-right: 10px;
  margin-bottom: 10px;
}

body {
  display: flex;
  flex-wrap: wrap;
  margin: 10px;
}
```

flexbox

## 4.LOGIKA STRONY W JAVA SCRIPT, WAŻNE!!!!

Na początek omówmy etapy, które składają się na funkcję, która zarządza czasem:

## 1. Pobieranie czasu i dnia tygodnia

```
let data = new Date();  
let hour = data.getHours();  
let m = new Date();  
let minute = m.getMinutes();
```

```
var day = new Date().getDay();  
// day = 1;
```

*zmienna hour, minute = pobiera czas z komputera klienta*

*w postaci liczby całkowitej (0-24, 0-60)*

*zmienna day pobiera dzień tygodnia z komputera klienta w postaci liczby całkowitej (0-6)*

## 2. Definiowanie co ma się wykonać o odpowiednim dniu tygodnia i godzinie

*instrukcja warunkowa switch-case określa jakie operacje wykonują się w danym dniu tygodnia*

```
switch (day) {  
  case 0: //Niedziela...  
  }  
  case 1: //Poniedziałek...  
  }  
  case 2: //Wtorek...  
  }  
  case 3: //Sroda...  
  }  
  case 4: //Czwartek...  
  }  
  case 5: //Piatek...  
  }  
  case 6: //Sobota...  
  }  
}
```

*Poniżej w case:1 znajdują się warunki, które muszą wykonać się w danym dniu, o danej godzinie* (każdy case posiada swoje, zdefiniowane w schemacie, który opisze później)

```
case 1: //Poniedziałek  
{  
  // alert('Poniedziałek');  
  document.getElementById('weekday').innerHTML = " Dzień: Poniedziałek";  
  if (hour == 8 && minute > 00 && minute < 45 || hour == 7 && minute <= 59) { //Lekcja1  
    document.getElementById('weekday').innerHTML = lekcja[15];  
    var roznica = (hour * 60 + minute) - (godziny[8] * 60 + minuty[50]);  
    var roznica = -roznica;  
    godzina1();  
    document.getElementById('next').innerHTML = 'Następna lekcja: ' + lekcja[8];  
    break;  
  } else if (hour == 8 && minute <= 50 || hour == 8 && minute > 50) { (local var) minuty: number[] : 35 }  
    var roznica = (hour * 60 + minute) - (godziny[9] * 60 + minuty[35]);  
    var roznica = -roznica;  
    Programowanie();  
    godzina2();  
    break;  
  } else if (hour == 9 && minute >= 35 || hour == 10 && minute < 25) { //Lekcja3  
    var roznica = (hour * 60 + minute) - (godziny[10] * 60 + minuty[25]);  
    var roznica = -roznica;  
    Programowanie();  
    godzina3();  
    break;  
  }  
}
```

**3.SCHEMAT, o który opiera się większość logiki strony, określa którego dnia o której godzinie wykona się następujący po sobie kod(wyświetli odpowiednie dane)**

funkcja, która odpowiada za wyświetlanie elementów w HTML zdefiniowana w osobnym pliku godziny.js

wzór, który oblicza ile minut zostało do końca aktualnej lekcji na podstawie zmiennych hour i minute

```
document.getElementById('weekdays').innerHTML = '';
if (hour == 8 && minute > 00 && minute < 45 || hour == 7 && minute <= 59) { //lekcja1
    document.getElementById('weekday').innerHTML = lekcja[15];
    var roznica = (hour * 60 + minute) - (godziny[8] * 60 + minuty[50]);
    var roznica = -roznica;
    godzina1();
    document.getElementById('next').innerHTML = 'Następna lekcja: ' + lekcja[8];
    break;
} else if (hour == 8 && minute <= 50 || hour == 8 && minute >= 45 || hour == 9 && minute < 35) { //lekcja2
    var roznica = (hour * 60 + minute) - (godziny[9] * 60 + minuty[35]);
    var roznica = -roznica;
    Programowanie();
    godzina2();
    break;
}
```

funkcja, która odpowiada za wyświetlanie elementów w HTML zdefiniowana w osobnym pliku przedmioty.js

warunek, o której godzinie muszą wyświetlić się dane informacje

## LOGIKA ODLICZANIA CZASU

**warunek odliczania czasu do końca lekcji**

```
//Odliczanie czasu do końca lekcji
if (roznica >= 60 && roznica < 120) { ...
} else if (roznica >= 120 && roznica < 180) {
    roznica = roznica - 120;
    roznica = "2 godz. " + roznica;
} else if (roznica >= 180 && roznica < 240) {
    roznica = roznica - 180;
    roznica = "3 godz. " + roznica;
} else if (roznica < 0 && roznica > -60) {
    // roznica = 'Jeszcze lekcja się nie zaczęła!';
} else if (roznica == undefined) {
    roznica = 'Brak lekcji! 0'
} else {
    // roznica = 'Jeszcze lekcja się nie zaczęła!';
    roznica = roznica;
}
```

**warunek odliczania czasu do końca przerwy**

```
//PRZERWA
if (hour == 8 && minute >= 45 && minute <= 49 == true) { //przerwa1
    minute = minute - 50;
    minute = -minute;
    var przerwa = minute;
    roznica = przerwa + ' min. do końca przerwy';
} else if (hour == 9 && minute >= 35 && minute <= 39 == true) { //przerwa2
    minute = minute - 40;
    minute = -minute;
    var przerwa = minute;
    roznica = przerwa + ' min. do końca przerwy';
} else if (hour == 10 && minute >= 25 && minute <= 39 == true) { //przerwa3
    minute = minute - 40;
    minute = -minute;
    var przerwa = minute;
    roznica = przerwa + ' min. do końca przerwy';
} else if (hour == 11 && minute >= 25 && minute <= 29 == true) { //przerwa4
```

## LOGIKA WYŚWIETLANIA PLANU NA CAŁY DZIEŃ

```
const newDay = () => {  
  
  document.getElementById('1h').innerHTML = '8:00 - 8:45';  
  document.getElementById('2h').innerHTML = '8:50 - 9:35';  
  document.getElementById('3h').innerHTML = '9:40 - 10:25';  
  document.getElementById('4h').innerHTML = '10:40 - 11:25';  
  document.getElementById('5h').innerHTML = '11:30 - 12:15';  
  document.getElementById('6h').innerHTML = '12:20 - 13:05';  
  document.getElementById('7h').innerHTML = '13:10 - 13:55';  
  document.getElementById('8h').innerHTML = '14:00 - 14:45';  
  
  var day = new Date().getDay();  
  // day = 1;  
  switch (day) {  
    case 0: //Niedziela  
    {  
      // alert('Niedziela');  
      document.getElementById('dzien').innerHTML = 'Niedziela';  
      document.getElementById('lekcja1').innerHTML = "1." + lekcja[18];  
      document.getElementById('lekcja2').innerHTML = "2." + lekcja[18];  
      document.getElementById('lekcja3').innerHTML = "3." + lekcja[18];  
      document.getElementById('lekcja4').innerHTML = "4." + lekcja[18];  
      document.getElementById('lekcja5').innerHTML = "5." + lekcja[18];  
      document.getElementById('lekcja6').innerHTML = "6." + lekcja[18];  
      document.getElementById('lekcja7').innerHTML = "7." + lekcja[18];  
      document.getElementById('lekcja8').innerHTML = "8." + lekcja[18];  
    }  
  }  
}
```

### Czwartek

1. Język angielski  
8:00 - 8:45

2. Język polski  
8:50 - 9:35

3. Język niemiecki  
9:40 - 10:25

4. Matematyka  
10:40 - 11:25

## 5. HOSTING I KWESTIE TECHNICZNE STRONY

Całą stronę hostuję na platformie GitHub Pages. Myślałem, żeby cały projekt napisać w PHP+MySQL, ale GitHub Pages nie obsługuje kodu wykonywanego po stronie serwera. Jednak moje rozwiązanie posiada kilka zalet:

-Serwis GitHub oferuje pełną synchronizację z systemem kontroli wersji Git (wynaleziony w 2005r. przez twórcę jądra Linuxa Linusa Torvaldsa), dzięki temu mogę jednocześnie utrzymywać historię kodu i poprzez synchronizację zdalnego repozytorium GitHub z poziomu terminala aktualizować stronę podpiętą pod moją prywatną domenę myzsoit.pl

-JavaScript wykonywany jest po stronie klienta przez co strona będzie działać na dowolnym hosting, który obsługuje jedynie usługę taką jak np. Apache albo GitHub Pages. Serwer nie musi mieć skonfigurowanego systemu MySQL i PHP.

-Wprowadzenie zmian w danych ogranicza się do zmiany wartości tablicy js, i zmiany zmiennej która wywołuje wartość z tablicy.



