

Zaawansowane Programowanie Webowe

Lab 2. JavaScript

Cel zajęć:

Poznanie podstaw wykorzystania języka JavaScript.

JavaScript ⁴

- dynamic computer programming language
- lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages
- an interpreted programming language with object-oriented capabilities
- Advantages:
 - Less server interaction – You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
 - Immediate feedback to the visitors – They don't have to wait for a page reload to see if they have forgotten to enter something.
 - Increased interactivity – You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
 - Richer interfaces – You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.
- Features:
 - ignores spaces, tabs, and newlines
 - semicolons are optional
 - JS is a case-sensitive language
 - untyped language - JavaScript variable can hold a value of any data type
- Usage - script tag

```
<script>
  // code....
</script>
```

 - tag placement:
 - in <head>...</head> section - If you need a script to run on some events like i.e. onclick

⁴ https://www.tutorialspoint.com/javascript/javascript_overview.htm

- in `<body>...</body>` section - If you need a script to run as the page loads so that the script generates content in the page. In this case, you would not have any function defined using JavaScript
- in an external file and then include in `<head>...</head>` section

Opis składni, zmiennych, pętli, obiektów:

<https://www.tutorialspoint.com/javascript/index.htm>

ECMAScript

The ECMAScript specification is a standardized specification of a JavaScript language.

The most important versions:

ECMAScript 6 (2015 / ES6)

- adds significant new syntax for writing complex applications, i.a.:
 - class declarations
`class Foo { ... }`
 - ES6 modules like
`import * as moduleName from "..."; export const Foo`
 - iterators and `for...of` loops
 - arrow function expression
`() => {...}`
 - `let` keyword for local declarations, `const` keyword for constant variable declarations,
 - promises - promise is an object that represents the eventual completion (or failure) of an asynchronous operation, and its resulting value

ECMAScript 8 (2017)

- `async` functions
- `Object.values/Object.entries`

Szczegóły:

<https://medium.com/@madasamy/javascript-brief-history-and-ecmascript-es6-es7-es8-features-673973394df4>

Zadania

1. Stwórz grę w kółko i krzyżyk z wykorzystaniem HTML, CSS oraz vanilla JavaScript. Gra ma umożliwiać graczemu grę na zmianę z komputerem oraz [5pkt]:
 - wybór markera
 - zaczęcie gry od nowa
 - informować o zwycięzcy

Dodatkowe 3 punkty za napisanie "logiki" gry komputera, która nie będzie jedynie wybierała losowego, wolnego pola.

Dodatkowe 2 punkty za użycie składni wprowadzonej w nowych standardach, m.in. arrow functions, `let/const`, klasy, `for ... of`, `object.entries`, etc.

Dodatkowe 2 punkty za dodanie CSS-owych animacji do gry.

Kod dodaj do repozytorium do folderu Lab2.