



JS

JavaScript

лекція 4

Масиви та об'єкти

Масиви, тип даних Array

- Способи створення масивів
 - [elem1, elem2, elem3, ...],
 - [],
 - new Array(elem1, elem2, elem3, ...),
 - new Array(elem1)

Властивості та методи екземплярів Array

- Властивості

- Index - псевдо властивість масиву створеного в результаті порівняння з рег. виразом

```
var src = "This is a test string.";
var re = /\w+/g;
var arr;
while ( (arr = re.exec(src)) != null ) {
    alert( arr.index + "-" + arr);
}
```
- Input - псевдо властивість масиву створеного в результаті порівняння з рег. виразом яке містить вхідну строку

```
arr = re.exec(src)
alert(arr.input)
```
- length - довжина масиву
 - Доступно для запису

Властивості та методи екземплярів Array

- Методи

- `push` - додавання значень в масив. Повертає нову довжину
- `shift` - удаляє елемент з індексом 0 і зміщує решту елементів на 1 вниз. Повертає нову довжину
- `join()` - об'єднує всі елементи в строку. Приймає символ за допомогою якого виконається об'єднання, інакше «`,`»
- `concat` - приєднує значення до масиву. Якщо передати масив то приєднаються значення з цього масиву
- `pop` - повертає останній елемент масиву і видаляє його з масиву. + змінює довжину масиву
- `unshift` - добавляє нові елементи в початок масиву. Повертає нову довжину
- `slice` - повертає новий масив обрізавши поточний по індексу. Приймає індекси початку і кінця(не обов'язково) обрізання
- `reverse` - змінює код масиву змінивши порядок елементів на протилежний
- `sort` - сортує масив. Приймає функцію сортування. Якщо параметр не передати то сортується по лексикографічному порядку
- `splice` - удаляє елементи. Приймає індекс з якого починати та кількість(не обов'язково)

Об'єкти в JS, тип даних Object

- Способи створення
 - {}
 - new Object(value)

Властивості та методи екземплярів Object

- Властивості
 - length- має значення 1
 - prototype - дозволяє додавати властивості

Властивості та методи екземплярів Object

- Методи

- `Object.assign()` - Створює новий об'єкт шляхом копіювання значень всіх власних перерахованих властивостей з одного або більше вихідних об'єктів в цільовий об'єкт.
- `Object.create()` - Створює новий об'єкт із зазначеними об'єктом прототипу і властивостями.
- `Object.defineProperty()` - Додає до об'єкта іменоване властивість, що описується переданим дескриптором.
- `Object.defineProperties()` - Додає до об'єкта іменовані властивості, описувані переданими дескрипторами.
- `Object.freeze()` - Заморожує об'єкт: інший код не може видалити або змінити ніяке властивість.
- `Object.getOwnPropertyDescriptor()` - Повертає дескриптор властивості для іменованої властивості об'єкта.
- `Object.getOwnPropertyNames()` - Повертає масив, що містить імена всіх переданих об'єкту власних перераховуються і неперечісляємих властивостей.
- `Object.getOwnPropertySymbols()` - Повертає масив всіх символічних властивостей, знайдених безпосередньо в переданому об'єкті.

Властивості та методи екземплярів Object

- Методи

- `Object.getPrototypeOf()` - Повертає прототип зазначеного об'єкта.
- `Object.is()` - Порівнює, чи є два значення однаковими
- `Object.isExtensible()` - Визначає, чи дозволено розширення об'єкта.
- `Object.isFrozen()` - Визначає, чи був об'єкт заморожений.
- `Object.isSealed()` - Визначає, чи є об'єкт запечатаним (sealed).
- `Object.keys()` - Повертає масив, що містить імена всіх перерахованих властивостей переданого об'єкта
- `Object.observe()` - Асинхронно спостерігає за змінами в об'єкті.
- `Object.preventExtensions()` - Запобігає будь-яке розширення об'єкта.
- `Object.seal()` - Запобігає видаленню властивостей об'єкта іншим кодом.
- `Object.setPrototypeOf()` - Встановлює прототип (тобто, внутрішня властивість `[[Prototype]]`)

Контекст виконання функції в якості методу об'єкта

- Змінна `this` в середині функції
 - `this` - це посилання на об'єкт котрий викликає код в даний момент
 - ```
var user = {
 name: 'John Smith',
 getName: function() {
 console.log(this.name);
 }
};
user.getName(); // John Smith
```
- Виклик ф-ції як функції (`this` - `window`)
- Виклик ф-ції з оператором `new` (`this` - посилання на щойно створений об'єкт)

# Розуміння класу-конструктора, оператор instanceof, властивість constructor

- instanceof - використовується для перевірки, чи належить об'єкт даному типу.
- constructor - Повертає посилання на функцію Object, що створила прототип екземпляру



# Функція як об'єкт. Методи call та apply

- call()
- apply()

# Механізм наслідуванн, prototype

- prototype - властивість котра дозволяє організувати об'єкти в цепочки так щоб не знайдені властивості шукались в іншому об'єкті
- \_\_proto\_\_
- hasOwnProperty(prop) - повертає true якщо властивість належить самому об'єкту



# Завдання

- Створити об'єкт user з загальними властивостями для всіх користувачів
- Створити об'єкт admin котрий наслідується від user та розширяє його.
- Виконати перебір і вивід в консоль властивостей admin пропустивши наслідуємі властивості