



JS

JavaScript

функції

лекція 3

Імена функцій, їх створення та виклик

- Функція - спеціальний тип об'єктів, що дозволяє формувати визначену логіку поведінки і обробки даних
- Декларативний стиль
 - ```
function square (number1, number2){
 return number1 * number2;
}
```
- Функціональний стиль
  - ```
var square = function (number1, number2){  
    return number1 * number2;  
}
```
- Динамічне створення
 - ```
var square = new Function('number1', 'number2', 'return number1 * number2;');
```
- Функції методи - задані як значення властивості об'єкта
- Виклик за допомогою apply і call
  - ```
var result = fun.apply(thisArg[, argsArray]);
```
 - ```
var result = fun.call(thisArg[, arg1[, arg2[, ...]]]);
```

# Область видимості змінних

- Ключове слово var
- Доступ до зовнішніх змінних



# Аргументи функцій, об'єкт arguments

- arguments - локальна змінна функції, котра містить самі аргументи і ссилку на саму функцію
- arguments.length - кількість аргументів
- arguments.callee - посилання на саму функцію

# Інструкція return

- return - повернення результату роботи функції
- Повертається undefined якщо return відсутній

# Замикання і функції IIFE

- Замикання - це функція з усіма зовнішніми змінними котрі їй доступні
- Immediately-invoked function expression (негайно викликаємі функції)
  - `(function () { })()`;
  - `(function () { })( )`;
  - Передача параметрів
  - Повертаєме значення
  - Патерн - шаблон «модуль»



# Рекурсивні функції

- Функція котра викликає сама себе
  - Приклад: вирахування факторіала числа
  - $n! = n * (n-1) * (n-2) * \dots * 1$  ( $5! = 5*4*3*2*1$ , тобто 120)
  - ```
function factorial(n) {  
    if (n == 0) {  
        return 1;  
    } else {  
        return n * factorial (n-1);  
    }  
}
```

Завдання

- Створити функцію `validateUser` котра на вхід прийматиме об'єкт `user`, проводитиме валідацію властивостей та повертатиме об'єкт з властивостями `status` та `message`
- Для валідації властивостей використовувати окремо створені функції