



JS

# JavaScript

інструкції

лекція 2

# Поняття інструкцій

- Інструкції - засоби реалізації логіки програми
- Об'єднання інструкцій, оператор “,”
- Блок інструкцій { }

# Умовні інструкції

## • If

```
var name = 'Андрій'

if (name == 'Андрій') {
  console.log('Привіт, Андрій!');
}
```

## • If/else

```
var name = 'Андрій'

if (name == 'Андрій') {
  console.log('Привіт, Андрій!');
} else {
  console.log('Привіт, невідомий користувач');
}
```

## • If/else if/else

```
var name = 'Андрій'

if (name == 'Андрій') {
  console.log('Привіт, Андрій!');
} else if (name == 'Дмитро'){
  console.log('Привіт, Дмитро!');
} else if (name == 'Василь') {
  console.log('Привіт, Василь!');
} else if (name == 'Микола'){
  console.log('Привіт, Микола!');
} else {
  console.log('Привіт, невідомий користувач');
}
```



# Інструкції циклів

- for

```
for (var i = 0; i < 9; i++) {  
    console.log(i);  
    // будь-які дії  
}
```

- for .. in ..

```
var obj = {a:1, b:2, c:3};  
  
for (var prop in obj) {  
    console.log("obj." + prop + " = " + obj[prop]);  
}
```

- while

```
var i = 0;  
do {  
    i += 1;  
    console.log(i);  
} while (i < 5);
```

- do .. while

```
var i = 0;  
do {  
    i += 1;  
    console.log(i);  
} while (i < 5);
```

# Інструкції переключення

- switch/case/default

```
var name = 'Андрій'
```

```
switch (name) {  
  case 'Андрій':  
    console.log('Привіт, Андрій!'); break;  
  case 'Дмитро':  
    console.log('Привіт, Дмитро!');  
    break;  
  case 'Василь':  
    console.log('Привіт, Василь!');  
    break;  
  case 'Коля':  
  case 'Микола':  
    console.log('Привіт, Микола!');  
    break;  
  default:  
    console.log('Привіт, невідомий користувач');  
}
```



# Інструкції переривання і повернення

- `return` – зупинка поточної функції, повернення результату
- `continue` – зупиняє поточну ітерацію циклу і передає управління до вершини циклу для наступної ітерації
- `break` – зупиняє поточний цикл і передає управління наступній інструкції
- `throw` – викликає виключення

# Інструкції перехоплення і обробки виключень

- try/catch/finally

```
try {  
    throw new Error("oops");  
}  
catch (ex) {  
    console.error("inner", ex.message);  
    throw ex;  
}  
finally {  
    console.log("finally");  
}
```

- Тип даних Error

```
try {  
    throw new EvalError('eval');  
} catch (e) {  
    if (e instanceof EvalError) {  
        alert("Ошибка при выполнении: " + e.message);  
    } else {  
        throw(e)  
    }  
}
```

- new Error([message[, fileName[, lineNumber]]])

**EvalError** - Ошибка при выполнении eval()

**RangeError** - Эта ошибка происходит, когда переданный параметр лежит за пределами допустимого диапазона

**ReferenceError** - Ошибка при разыменовании неверной ссылки

**SyntaxError** - Синтаксическая ошибка, которая произошла при разборе кода, переданного eval()

**TypeError** - Переменная или параметр неправильного типа

**URIError** - Ошибка происходит, когда encodeURIComponent() или decodeURI() вызваны с неправильными параметрами



# Побудова таблиці в циклі

- Використання елемента `style`
- Використання правил CSS



# Завдання

- Створити об'єкт `user` з властивостями `"name"`, `"email"`, `"age"`, `"isAdmin"`
- У властивість `"age"` записати значення за допомогою виразу `Math.round(Math.random() * 100 )`  
(розібратись що значить такий запис)
- За допомогою вивчених інструкцій організувати вивід в консоль віку користувача з правильним множником слова `"рік"`  
Наприклад:  
1 рік  
2 роки  
...  
6 років  
...
- Реалізувати завдання різними способами
- Проаналізувати та визначити який спосіб краще і чому