АО «АЛМАТИНСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»
https://atu.edu.kz/

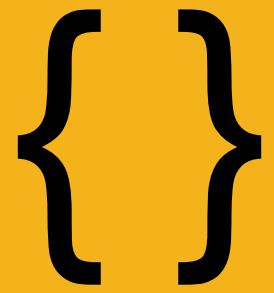
Факультет «Информационные технологии»
Кафедра «Компьютерная инженерия»

Название курса: «Разработка клиентских Web-приложений»



Зиятбекова Гулзат Зиятбеккызы – PhD, ассоциированный профессор
g.ziyatbekova@atu.edu.kz

JAVASCRIPT · ОСНОВЫ



Лекция № 12
Управляющие конструкции, функции и область видимости

if

выбор

for

повторение

ƒ

задача

Условия · циклы · функции · scope

КАРТА ТЕМЫ

Программа может выбирать и повторять

Управляющие конструкции задают направление выполнения JavaScript-кода.

if

Условия

Выбирают путь выполнения.

↻

Циклы

Повторяют действие.

ƒ

Функции

Собирают задачу для повторного вызова.

Сегодня: if...else, switch, for, while, функции, arrow functions и scope.

УСЛОВИЕ

if...else выбирает один из двух путей

ПРОВЕРКА

score >= 50

ДА · TRUE

«Пройдено»

Если условие истинно, выполняется первый блок.

НЕТ · FALSE

«Не пройдено»

Иначе выполняется блок else.

НЕСКОЛЬКО ВАРИАНТОВ

switch сравнивает конкретные варианты

Он подходит, когда одну переменную нужно сопоставить с несколькими значениями.

01

switch

Берём одну переменную.

02

case 1

day = 1 → «Понедельник».

03

case 2

day = 2 → «Вторник».

04

default

Неизвестный случай.

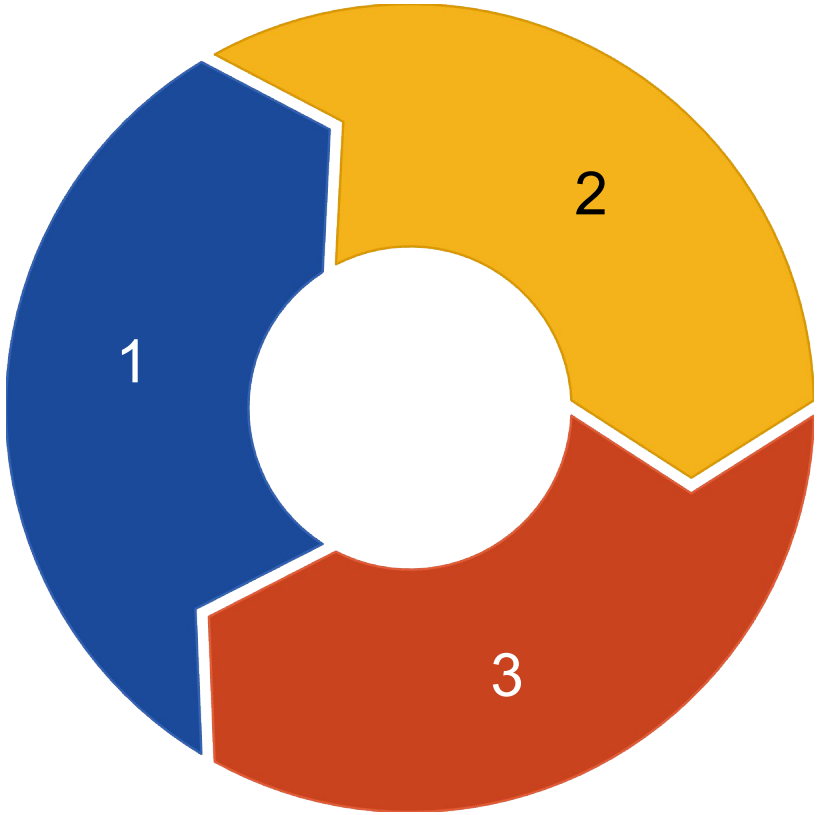
break завершает выполнение выбранной ветки.

ПОВТОРЕНИЕ

Циклы повторяют действие

Действие

Цикл выполняет одну операцию несколько раз.



Проверка

Повторение продолжается, пока выполняется условие.

Остановка

Важно заранее понимать, когда цикл должен остановиться.

ЦИКЛ СО СЧЁТЧИКОМ

for объединяет старт, условие и шаг

01

Инициализация

let i = 0

02

Проверка

i < 5

03

Тело цикла

Выводит текущее значение i.

04

Шаг

i++

РЕЗУЛЬТАТ

0, 1, 2, 3, 4

ЦИКЛ С УСЛОВИЕМ

while сначала проверяет условие

```
while (count < 5) {
  count++;
}
```

Проверка до тела. Если условие ложно с самого начала, тело не выполнится ни разу.

Проверка

count < 5

Тело

count++

Назад

Новая проверка.

Стоп

Условие ложно.

Изменяйте счётчик внутри цикла, чтобы избежать бесконечного повторения.

ПЕРЕИСПОЛЬЗОВАНИЕ

Функция — переиспользуемая задача

```
function sum(a, b) {
  return a + b;
}

sum(5, 3); // 8
```



Операция

Функция объединяет код определённой операции.



Параметры

a и b принимают входные значения.



return

Возвращает результат: 8.

СОВРЕМЕННЫЙ СИНТАКСИС

Arrow function записывает функцию короче

ОБЫЧНАЯ ФУНКЦИЯ

```
function sum(a, b) {
  return a + b;
}
```

ARROW-ВЕРСИЯ

```
const sum = (a, b) => a + b;
```

Слева — параметры.

Справа — неявный return.

Широко используется в современном JavaScript-коде.

ДОСТУПНОСТЬ ПЕРЕМЕННЫХ

Scope определяет доступность переменной

GLOBAL SCOPE

Внешний уровень

Переменная доступна в глобальной области.

FUNCTION SCOPE

Внутри функции

Переменная доступна внутри функции.

BLOCK SCOPE

Внутри блока { }

let и const имеют блочную область видимости.

```
let outside = 1;

function task() {
  const inside = 2;

  if (inside) {
    let local = 3;
  }
}
```