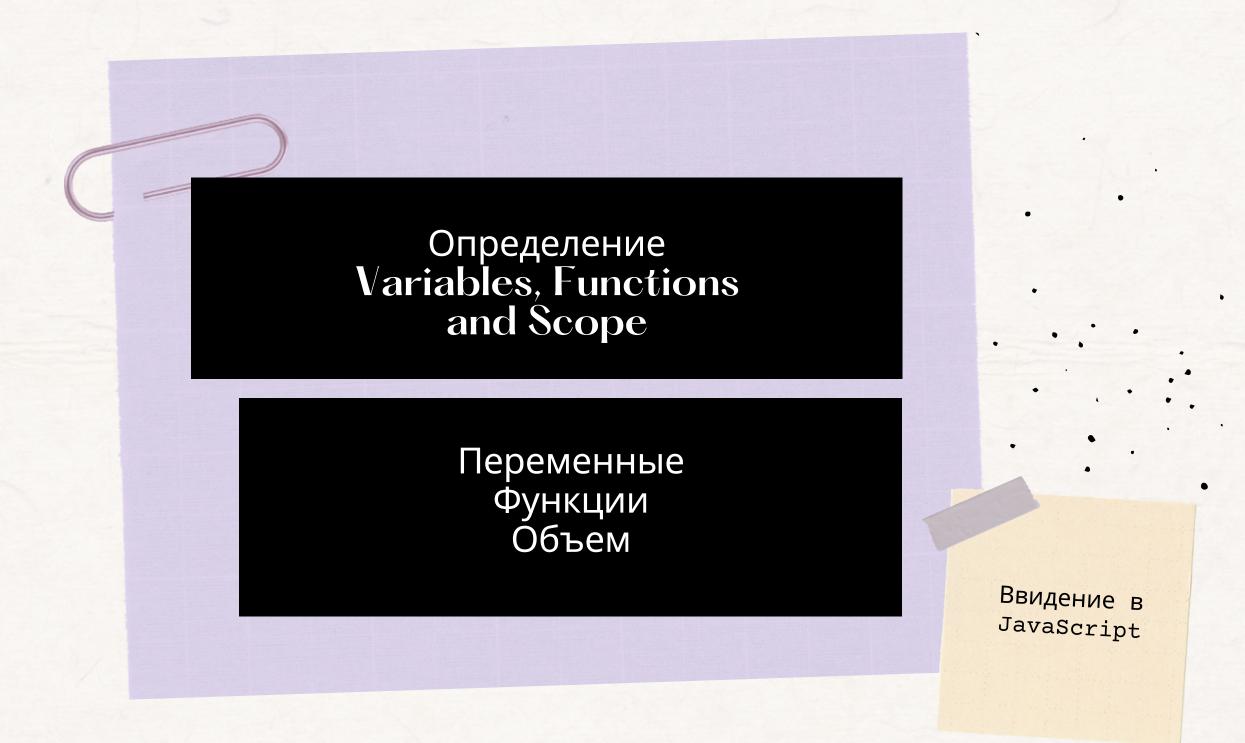
#### Assem Mukhanbet



## Variables –

## Переменные



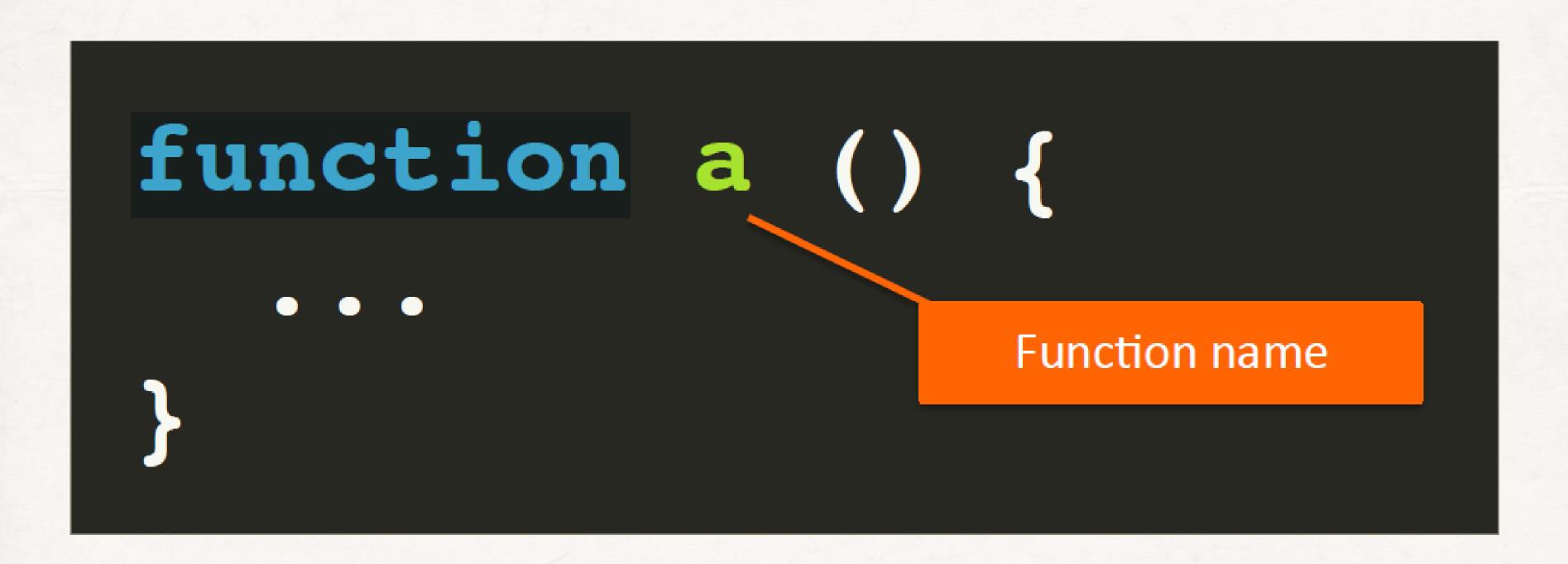
Определение переменной всегда должно начинаться с «var».



Типы не объявлены

- JS это динамически типизированный язык.
- Одна и та же переменная может содержать разные типы в течение всего времени выполнения.

Функции



Функции

Значение назначенной функции, <u>НЕ возвращаемый результат!</u>

Имя не определено

Функции

Выполняет функцию (она же вызываемая функция)

Функции

Аргументы, определенные без "var"

```
function compare (x, y) {
  return x > y;
}
```

Функции

```
function compare (x, y) {...}
var a = compare(4, 5);
compare(4, "a");
compare();
```

## Scope-

Объем

## Global

Переменные и определенные функции здесь доступны повсюду

# Function aka lexical

Переменные и функции, определенные здесь доступны только внутри этой функции

## Scope Chain— Цепочка областей

- ! Все выполняется в контексте выполнения ! Вызов функции создает новый контекст выполнения ! Каждый контекст выполнения имеет:
  - Собственная переменная среда
    - Специальный «this» объект
  - Ссылка на его внешнюю среду
- ! Глобальная область не имеет внешней среды, поскольку это самое внешнее

## Scope Chain— Цепочка областей

Ссылочная (не определенная) переменная будет сначала искатся в своей текущей области. Если не найдено, будет выполняться поиск по внешней ссылке. Если он не найден, будет выполняться поиск внешней ссылки внешней ссылки и т. д. Это будет продолжаться в глобальной области видимости. Если переменная не найдена в глобальной области видимости, она не определена.

## Global

```
var x = 2;
A();
```

#### **Function A**

```
var x = 5;
```

B();

Даже если «В» вызывается внутри «А»

«В» определяется в Global

#### **Function B**

console.log(x);

Результат х=2