



Передовые
инженерные
школы



МИНОБРНАУКИ
РОССИИ



УНИВЕРСИТЕТ
ИННОПОЛИС

Занятие 3

ОСНОВЫ



План занятия

1. Идентификаторы: переменные и константы
2. Типы данных
3. Коллекции: массивы и объекты

Идентификаторы

Рассмотрим пример кода и какие в нем есть идентификаторы

```
const MAX = 2000;
```

```
const calculateSubtotal = (goods) => {  
  let amount = 0;  
  for (const item of goods) {  
    if (item.price < 0) throw 'Negative price';  
    amount += item.price;  
  }  
  return amount;  
};
```

Идентификаторы

Переменная let

```
let amount = 0
```

let	- ключевое слово
amount	- название переменной
=	- символ присвоения
0	- значение переменной

Присвоение нового значения

```
amount = 50
```

Константа const

```
const MAX= 2000
```

const	- ключевое слово
MAX	- название константы
=	- символ присвоения
2000	- значение переменной

Идентификаторы

Локальная константа `const`

```
const total = 2000
```

<code>const</code>	- ключевое слово
<code>total</code>	- название константы
<code>=</code>	- символ присвоения
<code>2000</code>	- значение переменной

Локальные константы договоренность для обозначения промежуточных значений, будут находиться внутри функций или других конструкциях языка

Идентификаторы

Как не нужно объявлять переменные

~~let year = 22,
migration = 'Canada', I,
counter = 0~~

Идентификаторы

Правильно объявлять так переменные

```
let year = 22,  
let migration = 'Canada'  
let = 1  
let counter = 0
```


Идентификаторы

Выделим константы

```
const year = 22,  
const migration = 'Canada'  
let = 1  
let counter = 0
```

Идентификаторы

Промежуточные идентификаторы

```
const goods = trash[groupClass]  
const amount = calculateSubtotal(foods)  
expenses.set(groupClass, amount)
```

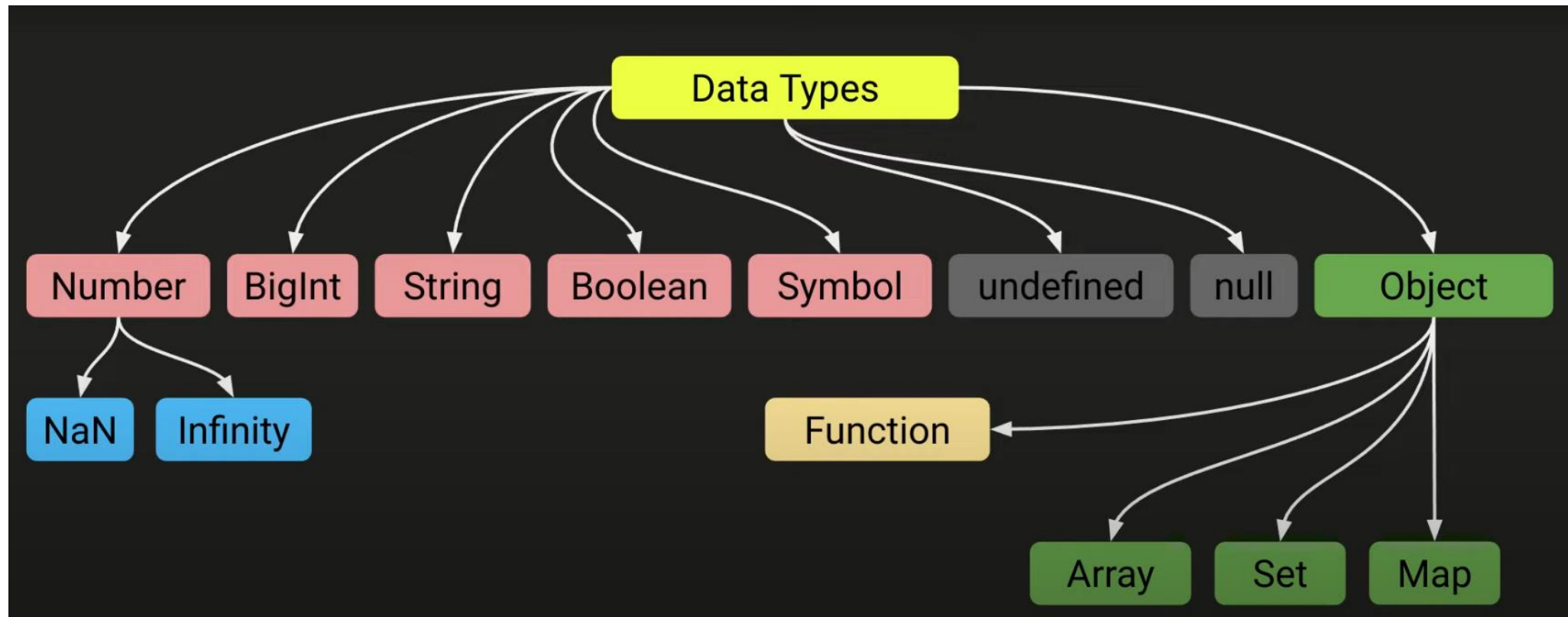
Идентификаторы

Без промежуточных переменных

```
expenses.set(  
    groupClass,  
    calcalateSubtotal(trash[groupClass])  
)
```

Типы данных

Классификация типов



Тип Number

```
const values = {  
  name: 'Mouse',  
  price: 25,  
  stock: false,  
  avg: (a, b) => (a + b) / 2n,  
  reference: null,  
  primitive: undefined,  
  numbers: [NaN, Infinity],  
};
```

Тип BigInt

```
const values = {  
  name: 'Mouse',  
  price: 25,  
  stock: false,  
  avg: (a, b) => (a + b) / 2n,  
  reference: null,  
  primitive: undefined,  
  numbers: [NaN, Infinity],  
};
```

Тип String

```
const values = {  
  name: 'Mouse',  
  price: 25,  
  stock: false,  
  avg: (a, b) => (a + b) / 2n,  
  reference: null,  
  primitive: undefined,  
  numbers: [NaN, Infinity],  
};
```

Тип Boolean

```
const values = {  
  name: 'Mouse',  
  price: 25,  
  stock: false,  
  avg: (a, b) => (a + b) / 2n,  
  reference: null,  
  primitive: undefined,  
  numbers: [NaN, Infinity],  
};
```


Тип undefine

```
const values = {  
  name: 'Mouse',  
  price: 25,  
  stock: false,  
  avg: (a, b) => (a + b) / 2n,  
  reference: null,  
  primitive: undefined,  
  numbers: [NaN, Infinity],  
};
```

Тип null

```
const values = {  
  name: 'Mouse',  
  price: 25,  
  stock: false,  
  avg: (a, b) => (a + b) / 2n,  
  reference: null,  
  primitive: undefined,  
  numbers: [NaN, Infinity],  
};
```

Типы данных

Тип Object

```
const values = {  
  name: 'Mouse',  
  price: 25,  
  stock: false,  
  avg: (a, b) => (a + b) / 2n,  
  reference: null,  
  primitive: undefined,  
  numbers: [NaN, Infinity],  
}
```

Тип Function

```
const values = {  
  name: 'Mouse',  
  price: 25,  
  stock: false,  
  avg: (a, b) => (a + b) / 2n,  
  reference: null,  
  primitive: undefined,  
  numbers: [NaN, Infinity],  
}
```

Тип Array

```
const values = {  
  name: 'Mouse',  
  price: 25,  
  stock: false,  
  avg: (a, b) => (a + b) / 2n,  
  reference: null,  
  primitive: undefined,  
  numbers: [NaN, Infinity],  
}
```

Определение типа

```
const values = {  
  name: 'Mouse',  
  price: 25,  
  stock: false,  
  avg: (a, b) => (a + b) / 2n,  
  reference: null,  
  primitive: undefined,  
  numbers:  
} [NaN, Infinity],
```

```
for (const key in values) {  
  console.log('Type: ', typeof values[key])  
}
```

Коллекции: массивы и объекты

Структура кода состоящая из массивов и объектов

```
const purchase = {  
  Electronics: [  
    { name: 'Laptop', price: 1500 },  
    { name: 'Mouse', price: 25 },  
    { name: 'Keyboard', price: 100 },  
    { name: 'HDMI cable', price: 10 },  
  ],  
  Textile: [  
    { name: 'Bag', price: 50 },  
    { name: 'Mouse pad', price: 5 },  
  ],  
}
```

Коллекции: массивы и объекты

Структура кода состоящая из массивов и объектов

```
const purchase = {  
  Electronics: [  
    { name: 'Laptop', price: 1500 },  
    { name: 'Mouse', price: 25 },  
    { name: 'Keyboard', price: 100 },  
    { name: 'HDMI cable', price: 10 },  
  ],  
  Textile: [  
    { name: 'Bag', price: 50 },  
    { name: 'Mouse pad', price: 5 },  
  ],  
}
```

```
purchase.Electronics  
purchase['Electronics']  
purchase.Electronics[0]  
purchase.Electronics[1].price  
purchase.Electronics.length
```


Коллекции: массивы и объекты

Получим ключи из данной структуры

```
const purchase = {  
  Electronics: [  
    { name: 'Laptop', price: 1500 },  
    { name: 'Mouse', price: 25 },  
    { name: 'Keyboard', price: 100 },  
    { name: 'HDMI cable', price: 10 },  
  ],  
  Textile: [  
    { name: 'Bag', price: 50 },  
    { name: 'Mouse pad', price: 5 },  
  ],  
}
```

```
const keys = Object.keys(purchase)  
console.log(keys)
```

Получим
['Electronics', 'Textile']

Коллекции: массивы и объекты

Получим ключи и значения из данной структуры

```
const purchase = {  
  Electronics: [  
    { name: 'Laptop', price: 1500 },  
    { name: 'Mouse', price: 25 },  
    { name: 'Keyboard', price: 100 },  
    { name: 'HDMI cable', price: 10 },  
  ],  
  Textile: [  
    { name: 'Bag', price: 50 },  
    { name: 'Mouse pad', price: 5 },  
  ],  
}
```

```
const entries= Object.entries(purchase)  
console.log(entries)
```

Получим

```
[  
  [ 'Electronics', [Array] ]  
  [ 'Textile', [Array] ]  
]
```

Коллекции: массивы и объекты

Копирование массива

```
const items = [...purchase.Electronics]  
console.log(items)
```

```
const items = [  
  { name: 'Laptop', price: 1500 },  
  { name: 'Mouse', price: 25 },  
  { name: 'Keyboard', price: 100 },  
  { name: 'HDMI cable', price: 10 },  
]
```

Коллекции: массивы и объекты

Объединение массивов

```
const items = [...purchase.Electronics, ...purchase.Textile]  
console.log(items)
```

```
const items = [  
  { name: 'Laptop', price: 1500 },  
  { name: 'Mouse', price: 25 },  
  { name: 'Keyboard', price: 100 },  
  { name: 'HDMI cable', price: 10 },  
  { name: 'Bag', price: 50 },  
  { name: 'Mouse pad', price: 5 }  
]
```

Коллекции: массивы и объекты

Добавление элемента в массив

```
purchase.Electronics.push({ name: 'Monitor', price: 500 })  
console.log(purchase.Electronics)
```

```
const items = [  
  { name: 'Laptop', price: 1500 },  
  { name: 'Mouse', price: 25 },  
  { name: 'Keyboard', price: 100 },  
  { name: 'HDMI cable', price: 10 },  
  { name: 'Bag', price: 50 },  
  { name: 'Mouse pad', price: 5 },  
  { name: 'Monitor', price: 500 }  
]
```

Коллекции: массивы и объекты

Удаление элемента в массива

```
purchase.Electronics.pop()  
console.log(purchase.Electronics)
```

```
const items = [  
  { name: 'Laptop', price: 1500 },  
  { name: 'Mouse', price: 25 },  
  { name: 'Keyboard', price: 100 },  
  { name: 'HDMI cable', price: 10 },  
  { name: 'Bag', price: 50 },  
  { name: 'Mouse pad', price: 5 },  
]
```

Коллекции: массивы и объекты

Поиск элемента в массива

```
const items = [  
  { name: 'Laptop', price: 1500 },  
  { name: 'Mouse', price: 25 },  
  { name: 'Keyboard', price: 100 },  
  { name: 'HDMI cable', price: 10 },  
  { name: 'Bag', price: 50 },  
  { name: 'Mouse pad', price: 5 },  
]  
  
const price = items.find((item) => item.price > 100)  
console.log({price})
```

Коллекции: массивы и объекты

Другие методы массива

https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Array



Передовые
инженерные
школы



МИНОБРНАУКИ
РОССИИ



УНИВЕРСИТЕТ
ИННОПОЛИС

Спасибо за внимание

