

# Основы синтаксиса, типы данных, условный оператор

Василий Петров  
Разработчик Python, JavaScript



# Василий Петров

О спикере:

- Стаж работы в ИТ более 25 лет
- Разрабатывал корпоративные приложения
- Руководил проектами и ИТ-подразделениями
- Руководил собственным бизнесом
- В настоящее время участвую в различных проектах с применением Python и JavaScript

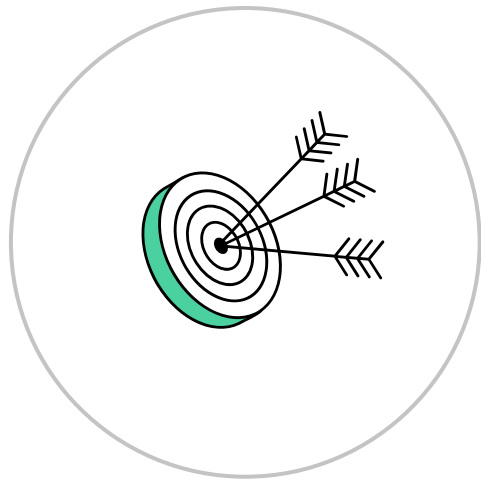




**Какие у вас ожидания  
от курса?**

# Цели занятия

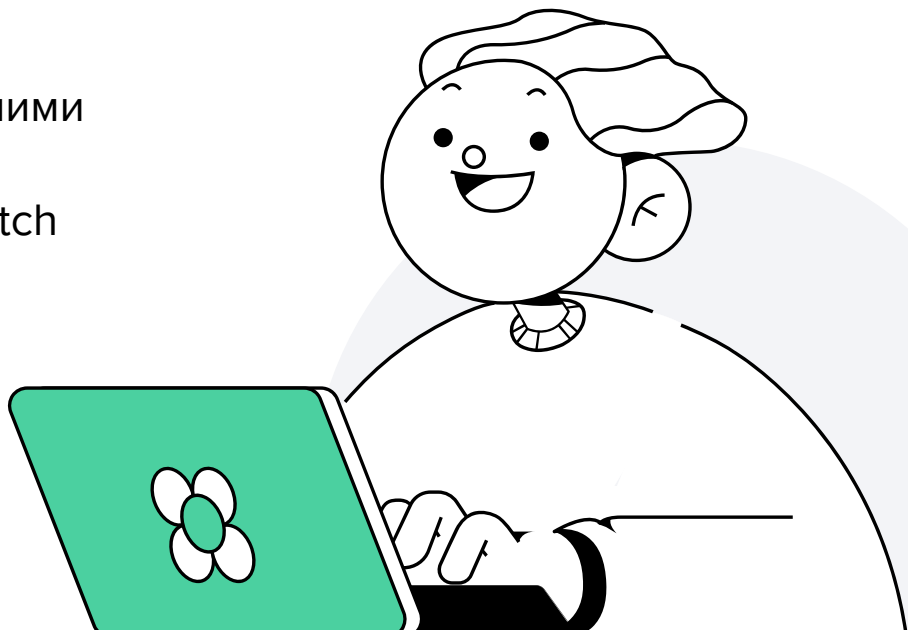
- Познакомимся с JavaScript и областью его применения
- Рассмотрим отличия JS от Python
- Рассмотрим базовый синтаксис и типы данных
- Напишем первую программу на JavaScript



# План занятия

- 1 История развития и области применения JS
- 2 Инструменты разработчика
- 3 Основы синтаксиса
- 4 Типы данных и основные операции с ними
- 5 Операторы. Условный, тернарный, switch
- 6 Итоги
- 7 Домашнее задание

\*Нажми на нужный раздел для перехода



# История развития и области применения JS



1

# История развития JS

1995	Разработан совместно компаниями Netscape и Sun Microsystems
1996	Аналог JScript от компании Microsoft
1997	Первая версия стандарта ECMAScript (ECMA-262)
конец 2000-х	Завершился длительный период различий в реализации в разных браузерах
2008	Google выпустил открытый движок V8 для встраивания в любые приложения
2009	Вышла первая версия платформы NodeJS
2012	Вышла первая версия языка TypeScript (расширение JS)
2015	Вышел стандарт ES6 (ES2015) — первые существенные изменения в языке
2015	Вышла первая версия IDE VS Code, полностью написанная на JS и TS
ежегодно	Выходят новые стандарты языка, последний — ES2021. Но изменений в них значительно меньше, чем в ES6 — ключевой момент, когда язык стал универсально применимым для больших проектов и различных платформ

# Вспомним

**Вопрос:** в каких областях применяют язык Python?





# Вспомним

**Вопрос:** в каких областях применяют язык Python?

**Ответ:**

- Обучение программированию
- Научное программирование
- Big Data
- Машинное обучение
- Серверный код
- Скрипты и утилиты



# Области применения JavaScript

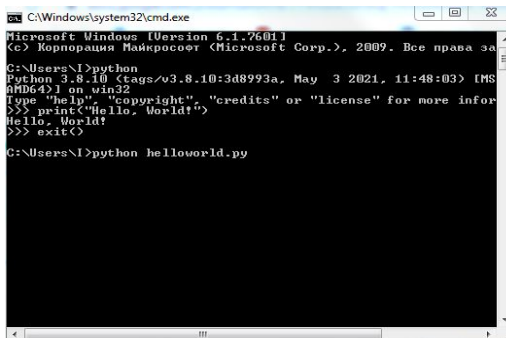
- Браузерные приложения
- Пользовательские интерфейсы для мобильных приложений
- Серверный код
- Приложения для десктопа
- Инфраструктурные утилиты для разработчиков



# Области применения JavaScript

## Python:

- интерпретатор из командной строки
- Jupyter Notebook



## JavaScript:

- браузеры
- NodeJS
- Встраиваемый движок JavaScript, например, V8



Chrome



Mozilla Firefox



nodeJS



v8

# Инструменты разработчика



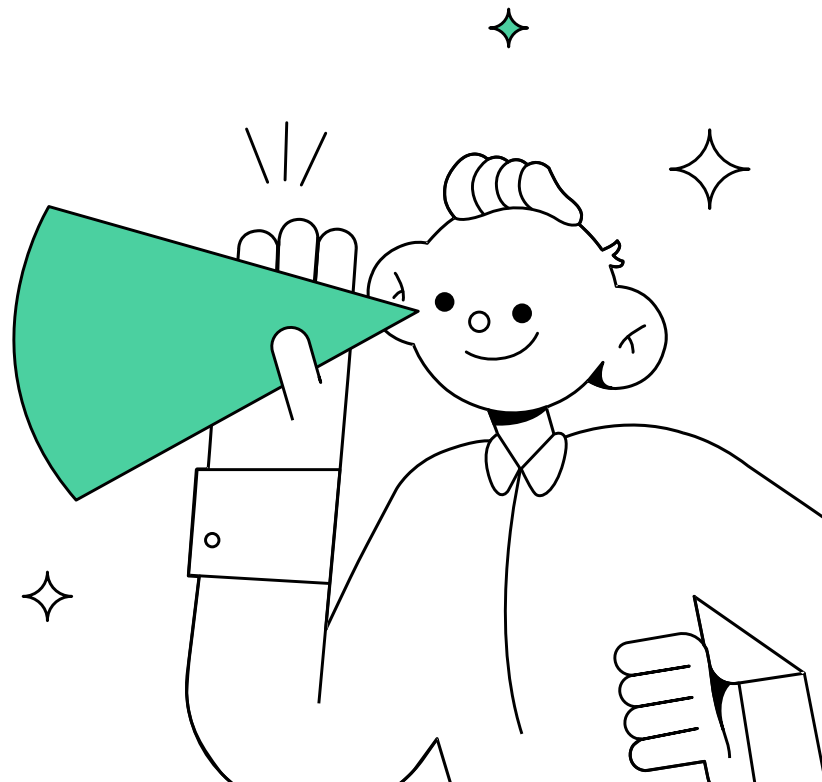
2

# Инструменты разработчика

Инструменты	Python	JavaScript
Текстовый редактор	✓	✓
IDE (VSCode)	✓	✓
IDE (PyCharm)	✓	
IDE (WebStorm)		✓
Python REPL	✓	
Блокноты Jupiter	✓	
DevTools в браузере		✓
NodeJS REPL		✓

# Демонстрация

DevTools в браузере Chrome,  
встраивание скрипта на JS в страницу  
(IDE VSCode)



# ОСНОВЫ синтаксиса



3

# ОСНОВЫ СИНТАКСИСА

## Python:

- отступы
- отсутствие блочных скобок
- один оператор в строке
- комментарии через #

```
def hello_world:  
    print("Hello, World!"); # prints greeting  
    for i in range(10): # prints even numbers  
        if i % 2 == 0:  
            print(i)
```

## JavaScript:

- необязательный разделитель — ; (необходим для нескольких операторов в строке)
- блочные скобки — {}
- нет жёстких требований по соблюдению отступов, но есть общепринятые
- несколько видов комментариев

```
function helloWorld() {  
    console.log("Hello, World!"); // prints greeting  
    /* prints even numbers */  
    for (i=0; i<10; i++) if (i%2 === 0) console.log(i) // bad style!!!  
}
```



# Идентификаторы и описание переменных

Идентификаторы и описание переменных	Python	JavaScript
Стиль написания составных идентификаторов	snake_case	camelCase
Специальные символы	допустим только _	допустимы _ и \$
Ключевые слова для определения переменных	нет	<b>var</b> и <b>let</b>
Ключевые слова для определения констант	нет	<b>const</b>
Модификаторы области видимости	<b>local</b> и <b>global</b>	нет модификаторов области видимости, но есть <b>hoisting</b> для <b>var</b> и <b>function</b>

# Типы данных и основные операции с ними



4

# Базовые типы данных

## Python:

- числовой тип неограниченной длины
- строки в Unicode (Python3)
- константы **True** и **False**
- тип `None`
- списки и словари
- логические операторы **and**, **or**, **not**
- операторы сравнения `==`, `!=`
- форматные строки `f'Hello, {name}!'`

## JavaScript:

- числовые типы ограниченной длины (но есть `BigInt`)
- строки в Unicode
- константы **true** и **false**
- два вида «пустоты» — **undefined** и **null**
- массивы и объекты
- логические операторы **&&**, **||**, **!**
- операторы сравнения `==` и `===`, `!=` и `!==` (неявные преобразования типов)
- шаблонные строки ``Hello, ${name}!``

# Демонстрация

типов данных и базовых  
операторов



# Списки и массивы

## Python:

- списки имеют определенную длину, обращение за пределы — ошибка
- разновидность списков — кортежи (неизменяемые списки)

## JavaScript:

- массивы имеют длину, но обращение за пределы возвращает **undefined**
- массив может иметь дыры
- массив является объектом и может также содержать свойства и методы

# Словари и объекты

## Python:

- обращение к ключам словарей через []
- обращение к несуществующему ключу — ошибка
- ключи могут быть любого неизменяемого типа
- всё в Python является объектом, но не словарём (класс Dict)

## JavaScript:

- обращение к свойствам объектов через . или []
- можно обращаться к несуществующим ключам — **undefined**
- ключи могут быть только строками
- всё в JS является объектом

# Другие типы данных

## Python:

- множества и операции с ними
- сложные структуры данных строятся на классах
- развитая система дополнительных структур данных во встроенных модулях
- регулярные выражения через отдельный модуль `re`

## JavaScript:

- типы `Set`, `HashMap`, `BigInt`, `ArrayBuffer`, `Buffer` и т. п.
- все сложные структуры данных строятся на объектах и классах (прототипы)
- встроенный тип `RegExp` для регулярных выражений

# Демонстрация

работы с массивами и объектами





# Операторы

Условный, тернарный, switch



5

# Условный оператор

## Python:

```
if a == b or a == c:  
    pass  
elif b==c:  
    pass  
else:  
    pass
```

## JavaScript:

```
if (a===b || a===c) {  
} else if (b===c) {  
} else {  
}
```

Обязательны скобки в условии. Блочные скобки не всегда обязательны, но с ними лучше читаемость кода

# Тернарный оператор

## Python:

```
a = 1 if c == d else 0
```

## JavaScript:

```
a = (c===d)?1:0.
```

Синтаксис унаследован от языков C и Java

# Оператор множественного выбора

## Python:

Отсутствует, необходимо написать цепочку **if-elif-else** либо действовать через структуры данных

```
if a == 0:  
elif a == 1:  
else:
```

## JavaScript:

Оператор **switch**

```
switch (a) {  
  case 0: break;  
  case 1: break;  
  default: break;  
}
```

# Напишем игру «Орёл/Решка» для браузера



# Итоги

Сегодня мы:

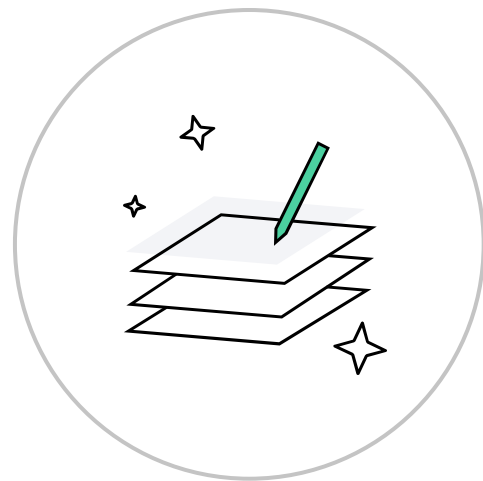
- 1 узнали, что такое JavaScript и какова его область применения
- 2 рассмотрели отличия JS от Python в базовом синтаксисе и основных типах данных
- 3 рассмотрели условный и тернарный оператор, а также оператор switch
- 4 попробовали написать первую программу на JavaScript для запуска в браузере



# Домашнее задание

Давайте посмотрим ваше домашнее задание.

- 1 Вопросы по домашней работе задавайте в чате группы
- 2 Задачи можно сдавать по частям
- 3 Зачёт по домашней работе ставят после того, как приняты все задачи



# Дополнительные материалы

- [Серия книг](#) «Вы не знаете JavaScript»
- [Современный учебник](#) JavaScript
- [Ресурсы](#) для разработчиков MDN





# Задавайте вопросы и пишите отзыв о лекции

Василий Петров  
Разработчик Python, JavaScript

