



# Control work, lesson 2

## *Control questions for today's lesson*

1. Какие есть подходы исполнения программного кода? Распишите преимущества и недостатки каждого подхода.
2. Какие есть подходы к типизации кода? Распишите преимущества и недостатки каждого подхода.
3. Какие типы данных есть в JS?
4. Какие типы примитивов есть в JS?
5. Как хранятся данные в памяти в каждом типе данных?
6. Чем процесс отличается от программы?
7. Что такое стек и куча? В чем отличие?

## **Answers**

1. Компиляция и интерпретация.

**Компиляция** — программный код переводится в машинные инструкции и запихивается в исполняемый файл, который идет на выполнение процессору, и который процессор понимает.

- Преимущества — высокая производительность, независимость от исходного кода и эффективное использование ресурсов.
- Недостаток — может потребоваться перекомпиляция, если нужно будет перенести код.

**Интерпретация** — программа выполняется построчно. Интерпретатор читает код программы и, спускаясь вниз по коду, последовательно выполняет инструкции.

- Преимущества — простота разработки, высокая переносимость и динамическая типизация.
  - Недостатки — зависимость от интерпретатора и низкая производительность (здесь можно вспомнить сравнение скорости выполнения различных вычислительных программ на плюсах и на питоне).
2. Существуют строго-тиปизированные и не строго типизированные языки. Их отличие заключается в том, что в нестрого типизированных языках компилятор динамически определяет, какой тип данных используется, то есть необязательно указывать тип данных, а в строго-типизированных нужно **обязательно** указывать тип данных. Например, C++ и C# — строго-типизированные языки, а JavaScript и Python — нет. Преимущества строго-типизированного языка — стандартизированность и, как следствие, большая безопасность кода. Плюсом нестрого типизированного является являющаяся быстрота разработки.
3. Типов данных в java скрипте 8: 7 из них — примитивы, оставшийся тип — объекты.
4. Типы примитивов:
- number — числовой тип
  - boolean — логический тип
  - string — строковый тип
  - null — пустота, этот тип означает, что в данной переменной ничего нет
  - undefined — неопределенный тип данных
  - bigint — для больших числе (больший  $2^{**53}$ )

- symbol — символьный тип данных.
5. Примитивы хранят какие-то данные. Процессор выделяет им определенное количество памяти в ячейках с определенным адресом. Объекты же хранят ссылки на хранящихся в них примитивов.
  6. *Процесс* — экземпляр программы, который создается при ее выполнении и которому операционная система выделяет ресурсы: процессорное время и оперативную память. *Программа* — это набор инструкций с определенным расширением.
  7. *Стэк* — область память, которая использует простую структуру для записи и чтения данных, что делает его очень быстрым, а также работает по алгоритму планирования Last In First Out (LIFO). Стэк применяется для работы с функциями.

*Xup* — медленная область оперативной памяти, которая выделяется для динамических объектов: списков, объектов, очередей и проч.