Pumpkin Spice Latte

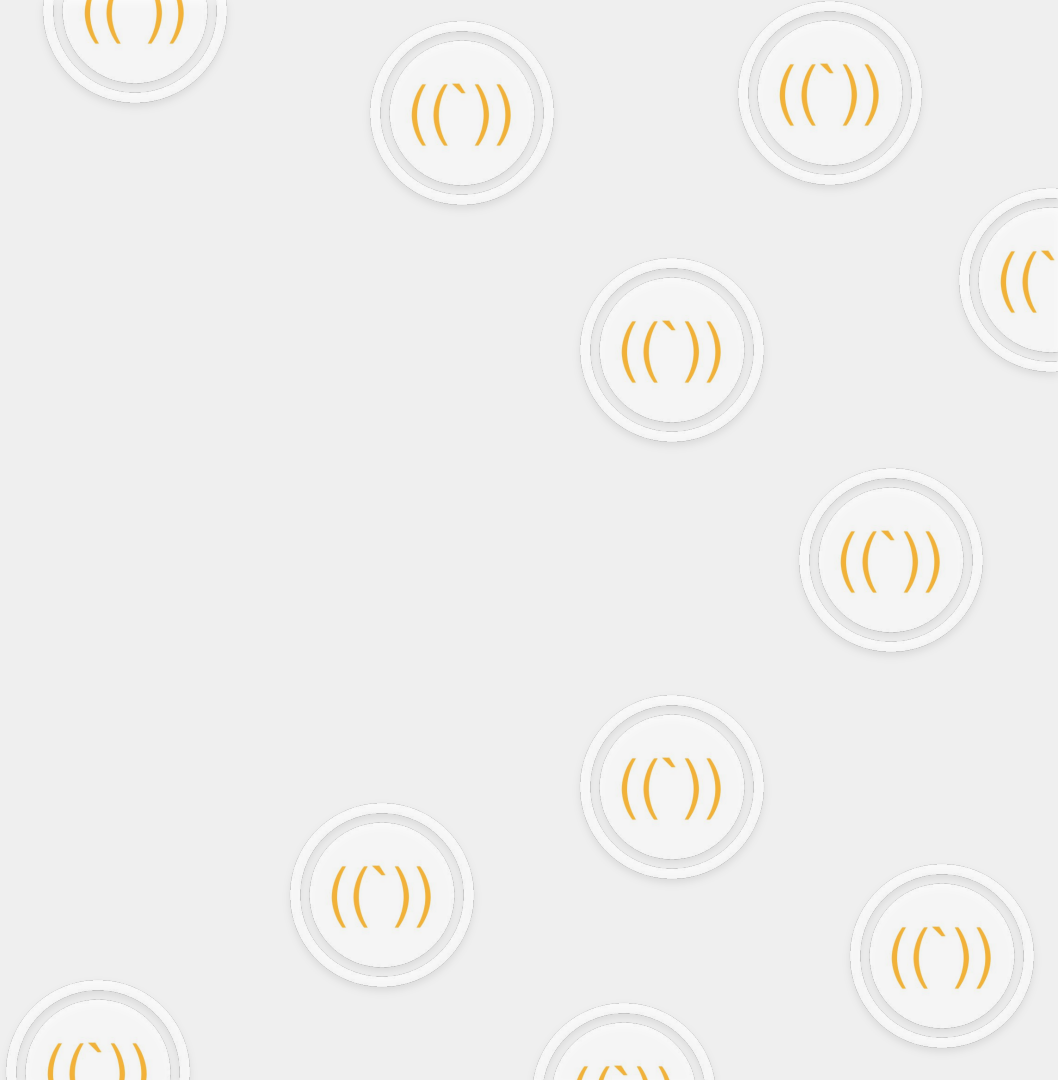
Beginner Server Programming

Overview

Simple HTTP:

- Python
- Ruby
- Java
- Rust
- Node.js
- C++

Python



Python

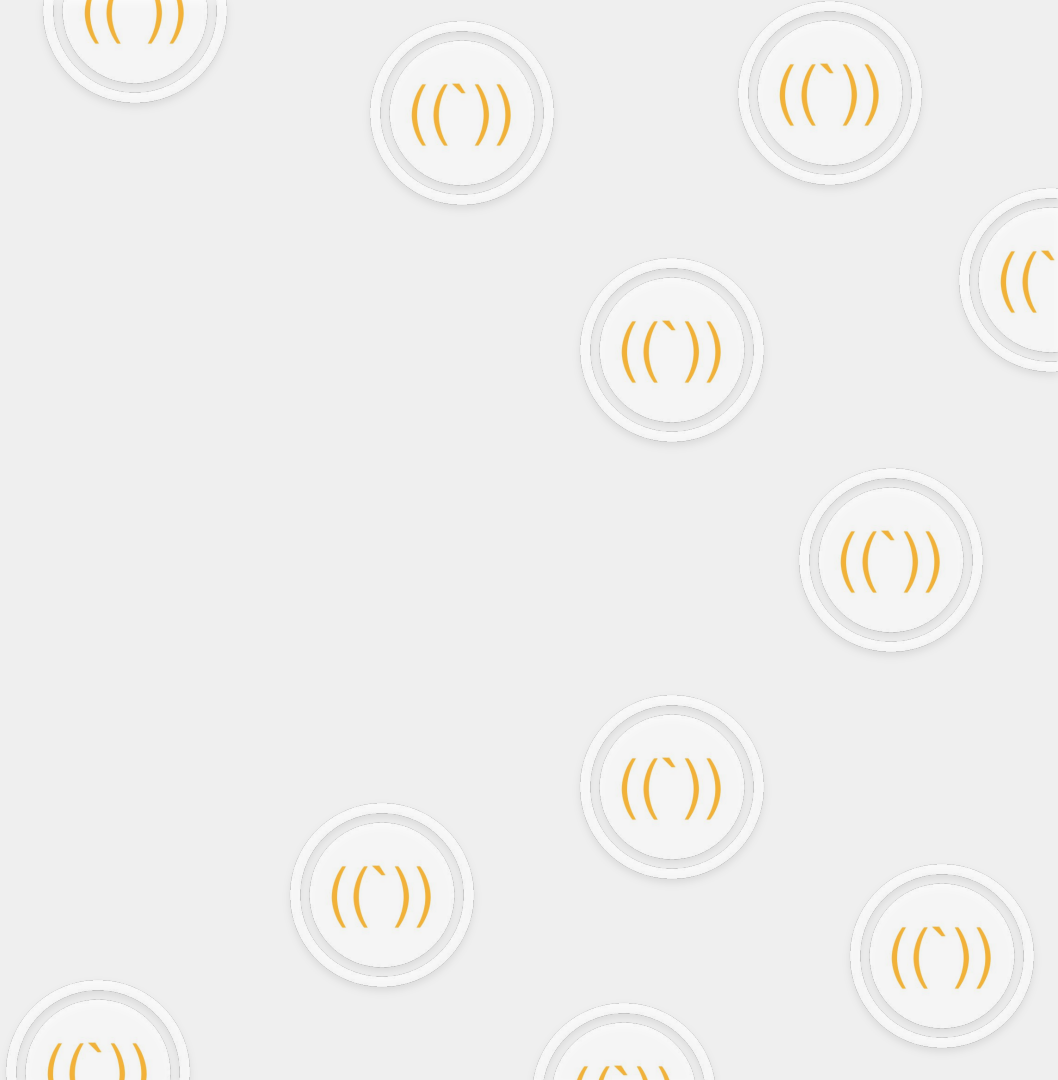
- Python is an interpreted high-level general-purpose programming language.
- Python's design philosophy emphasizes code readability with its notable use of significant indentation.
- It was introduced in February 1991.
- There are 2 ways to create a simple HTTP server using Python via:
  - your computer’s terminal
  - the interpreter using a **.py** extended file

Example

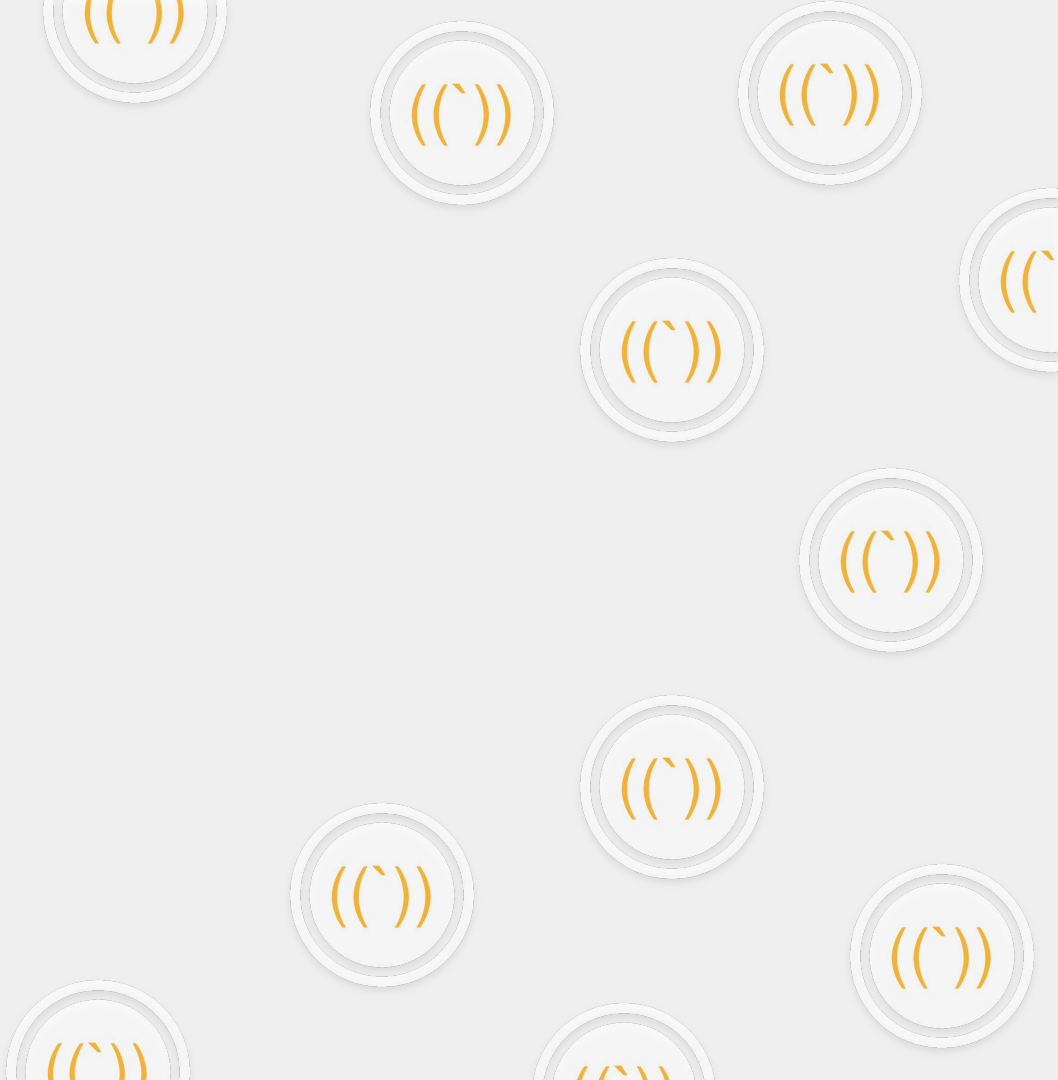
```
# Command Line / Terminal

$ cd ~/webprogramming/basicweb
$ python -m SimpleHTTPServer

# Serving HTTP on 0.0.0.0 port 8000 ...
# URL - http://127.0.0.1:8000
```

Tutorial

Python



Ruby



Ruby

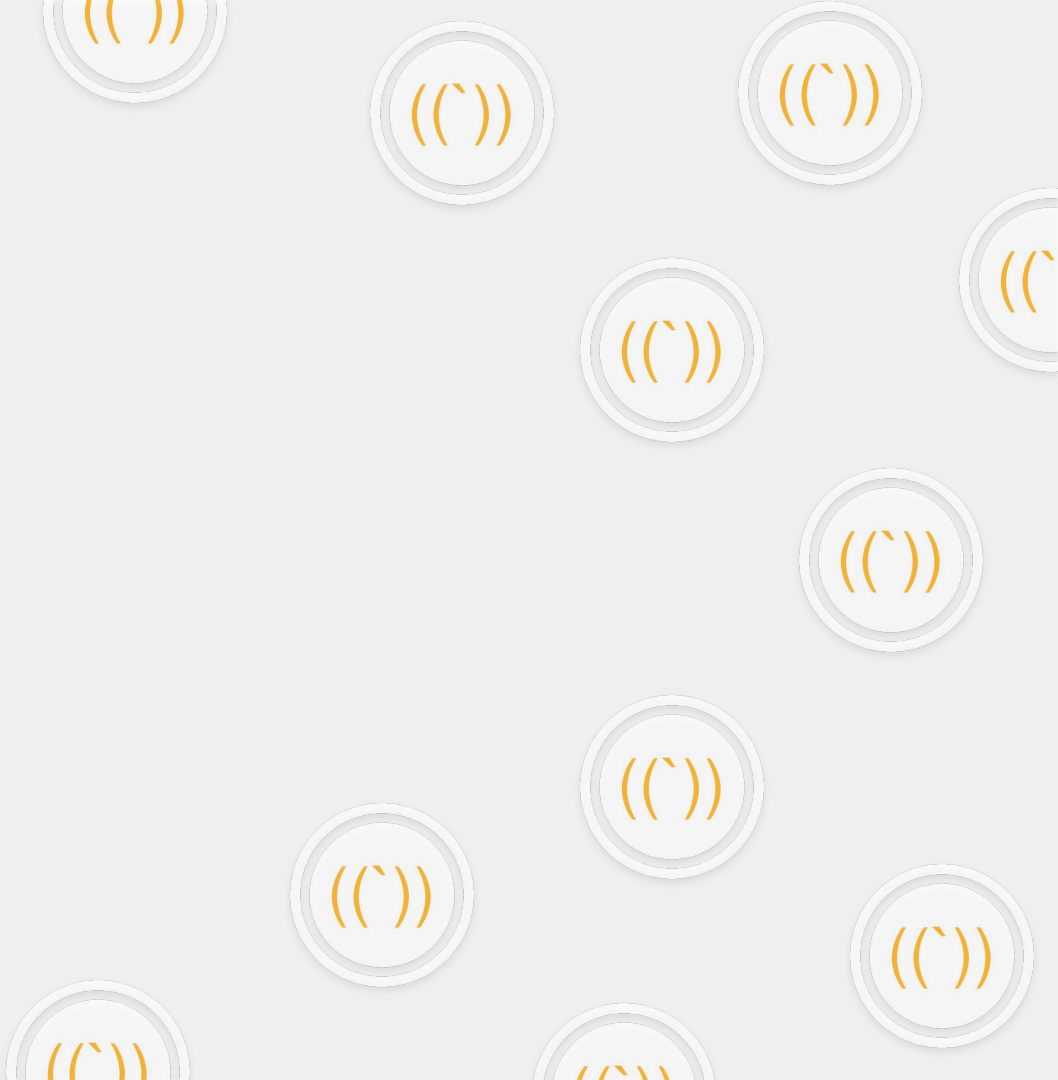
- Ruby is an interpreted, high-level, general-purpose programming language. It was designed and developed in the mid-1990s by Yukihiro "Matz" Matsumoto in Japan.
- Ruby is dynamically typed and uses garbage collection.
- There are 2 ways to create a simple HTTP server using Ruby via:
  - your computer's terminal
  - the interpreter using a **.rb** extended file

Example:

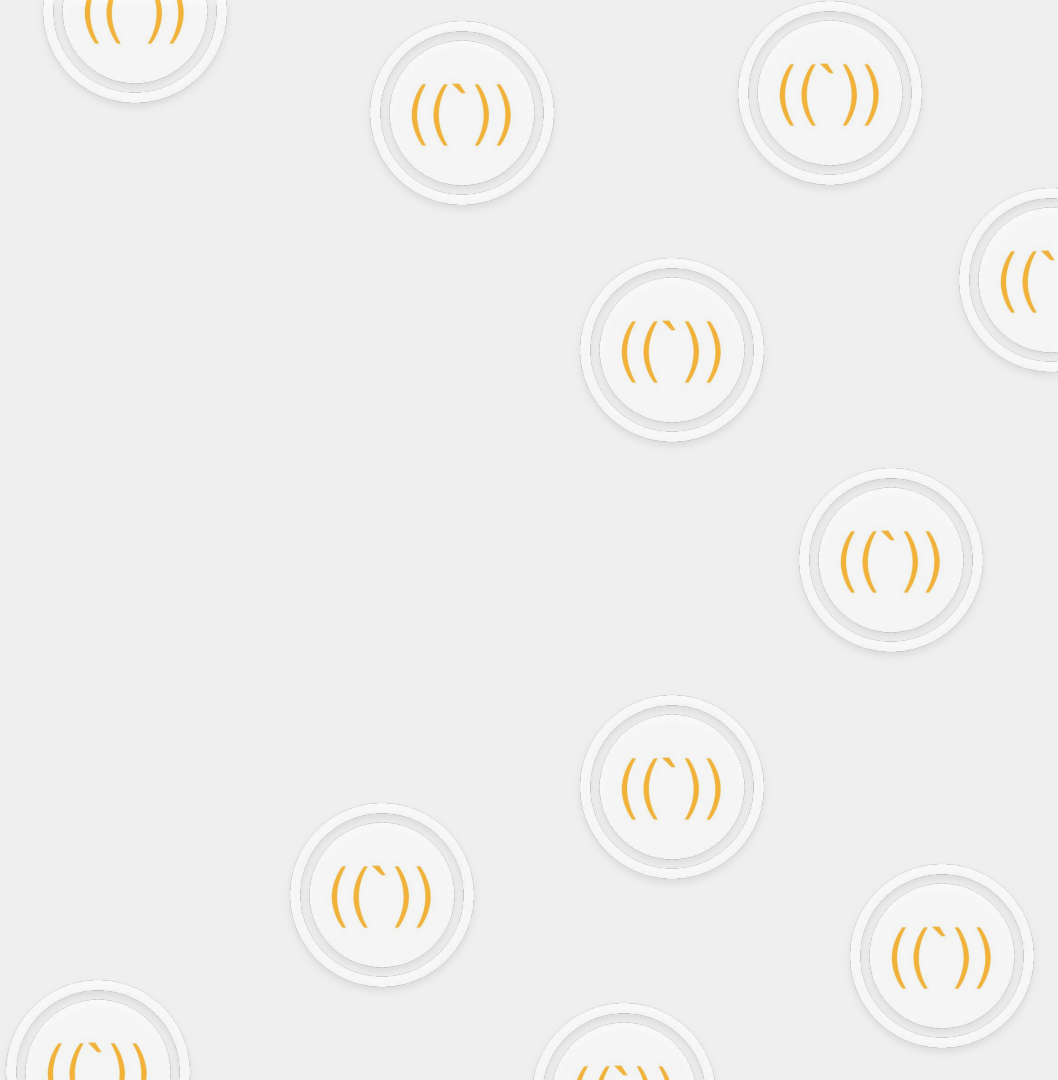
```
# Command Line / Terminal

$ cd ~/webprogramming/basicweb
$ ruby -run -e httpd . -p 8000

# URL - http://localhost:8000
```

Tutorial

Ruby



Node.js



Node.js

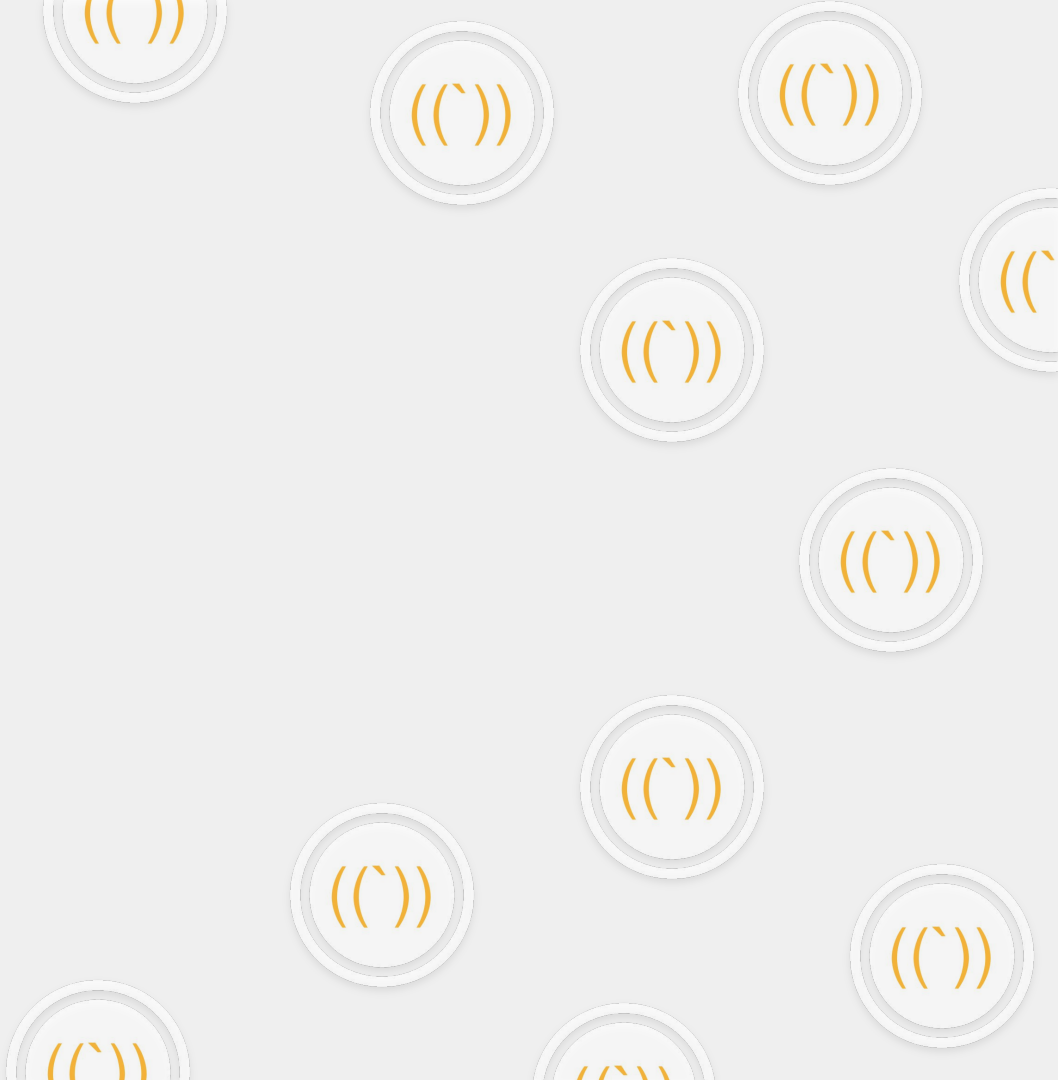
- Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser.
- It was introduced on May 27, 2009.
- Node.js does not require that you write the server in a file because it has a **shell**.
- It is much easier to debug and correct mistakes using an external file.
- The interpreter using a **.js** extended file to process and run code.

Example:

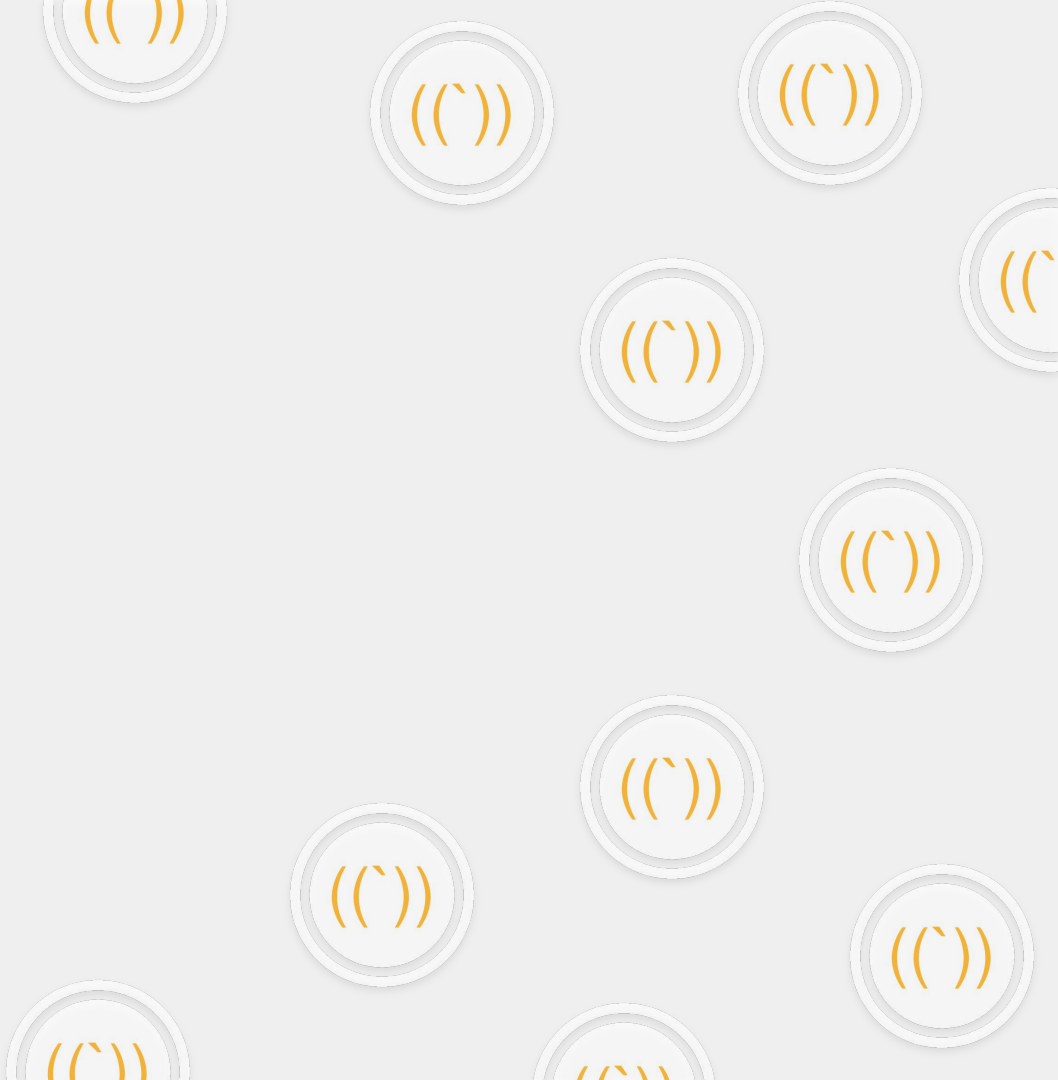
```
var http = require('http');

http.createServer(
  function (req, res) {
    res.write('Hello World!');
    res.end();
  }
).listen(8080);
```

Tutorial

Node.js



Java



Java

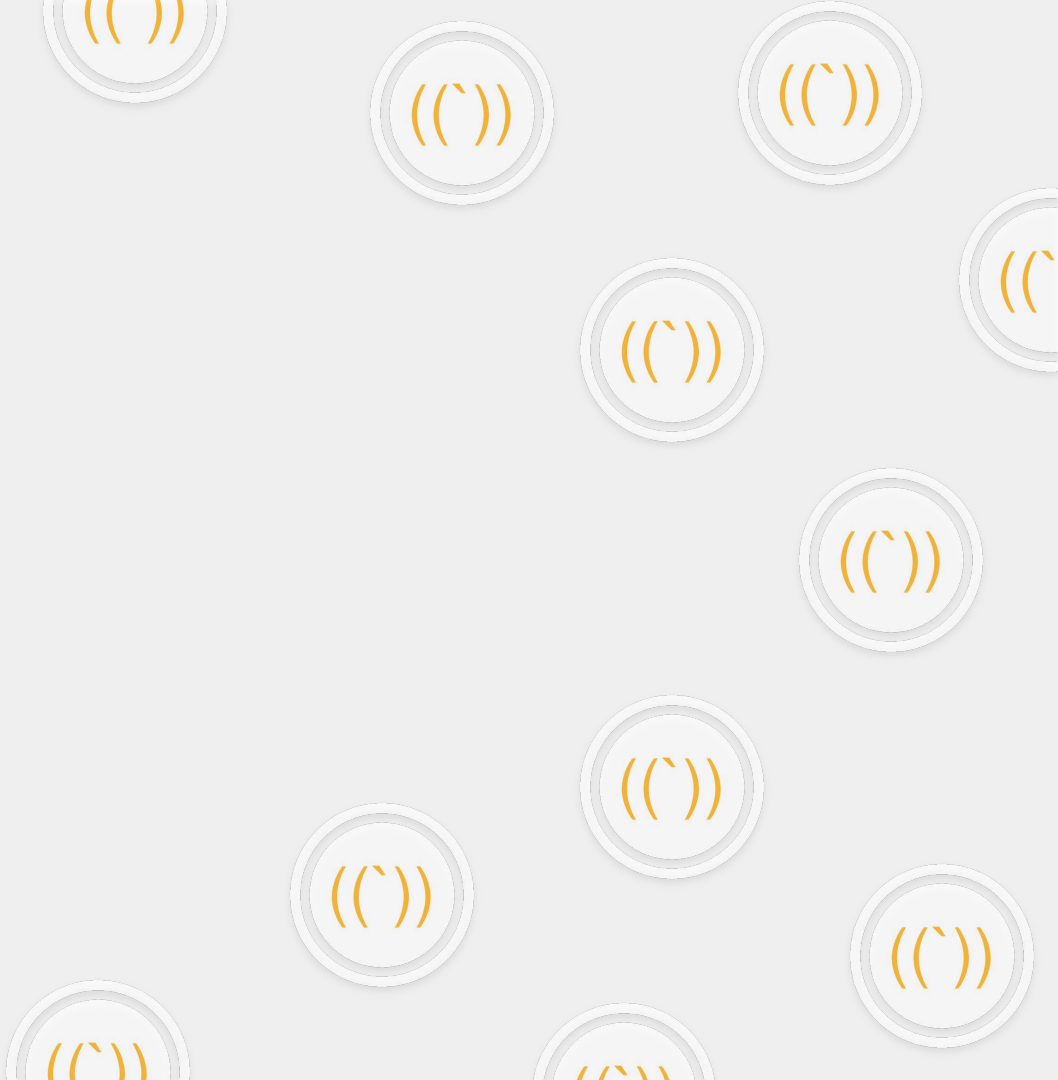
- Java is a High Level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible.
- It was introduced on May 23, 1995.
- Java servers require that they be written in an external file.
- The file name of your server is required to contain the same name as the public class that is defined in the file.
- The extension is **.java**:
  - Test.java

Example:

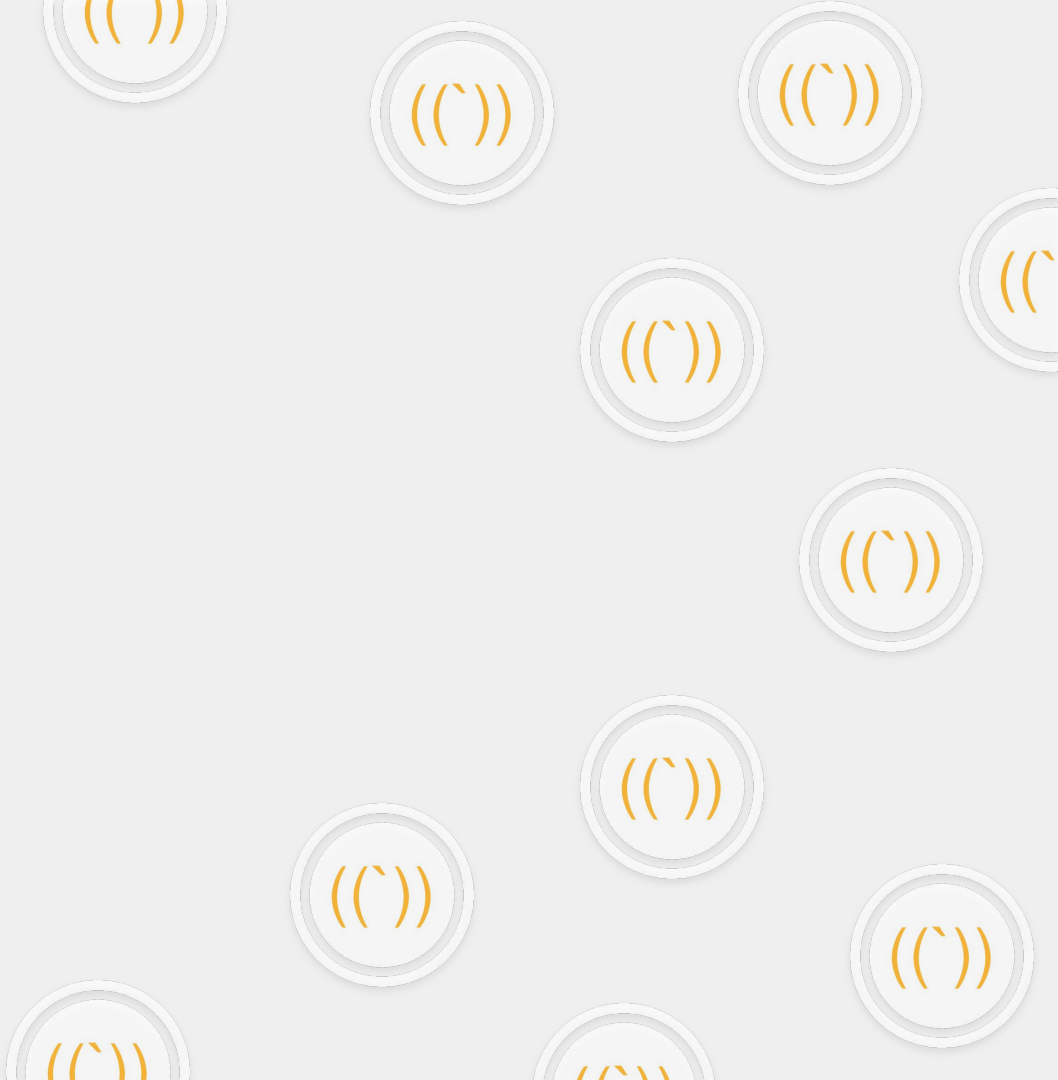
```
public class Test {
  public static void main(String[] args) throws
  Exception {
      HttpServer server = HttpServer.create(
        new InetSocketAddress(8000), 0
      );

      server.createContext(
        "/test", new MyHandler()
      );

      server.setExecutor(null);
      server.start();
  }
}
```

Tutorial

Java



Rust



Rust

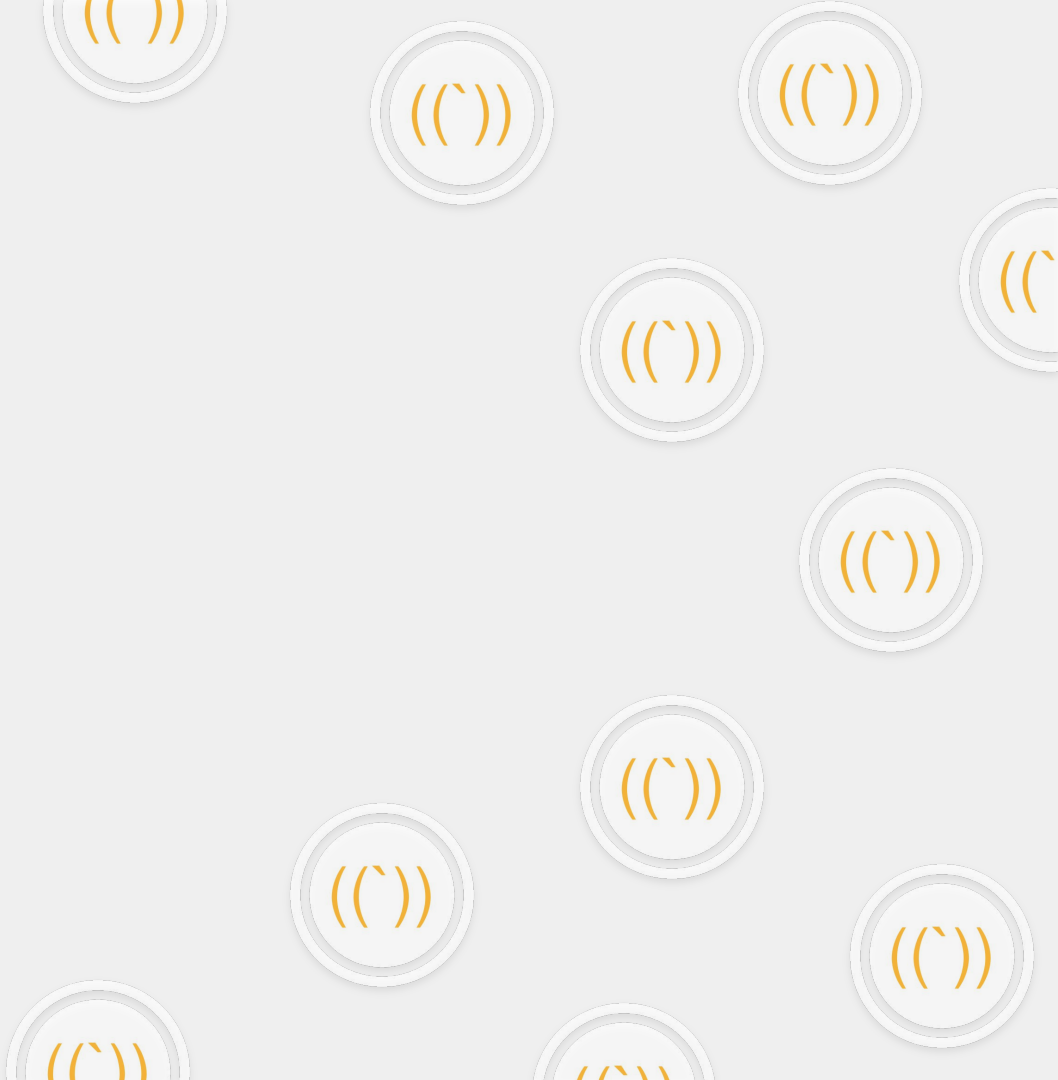
- Rust is a multi-paradigm programming language designed for performance and safety, especially safe concurrency.
- Rust is syntactically similar to C++, but can guarantee memory safety by using a borrow checker to validate references.
- It was introduced on July 7, 2010.
- Rust files typically are appended with **.rs** extension.

Example:

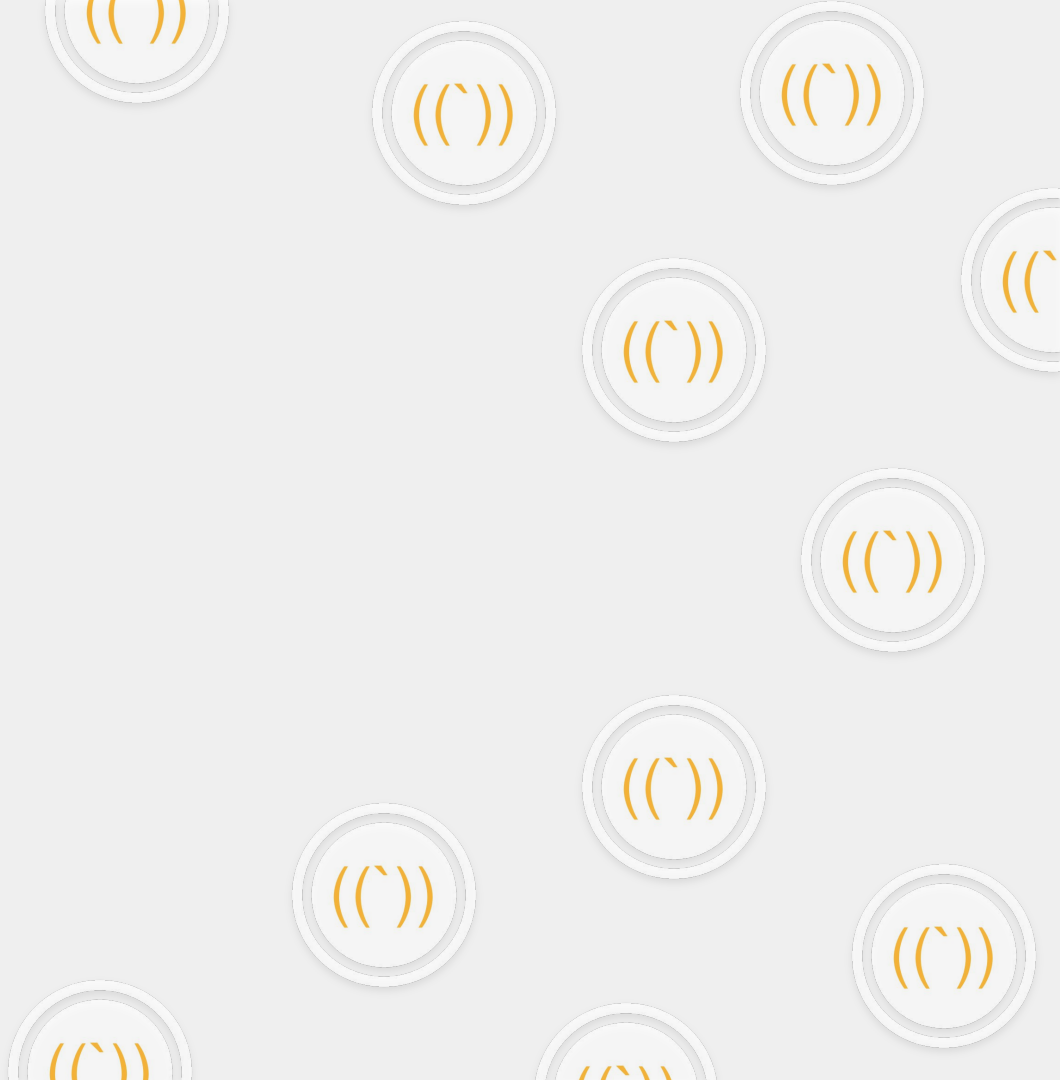
```
fn main() {
   // define listener
   for stream in listener.incoming() {
      match stream {
         Ok(stream) => {
            thread::spawn(|| {
               handle_client(stream)
            });
         }
         Err(e) => {
            println!("Unable to connect: {}", e);
         }
      }
   }
}
```

Tutorial

Rust



C++



C++

- C++ is a general-purpose programming language created by Bjarne Stroustrup as an extension of the C programming language, or "C with Classes".
- It was first introduced in 1985, but standardized in 1998.
- This is a compiled language, so it requires you to implement a server using external files.
- File extensions:
  - .c, .C, .cc, .cpp, .cxx, .c++
  - .h, .H, .hh, .hpp, .hxx, .h++

Example:

```
// this one is mega difficult compared to all of the
// other options

// Accept new TCP connection
// check if port is allowed

// Set IP address and port for server
// Allocate a socket

// Bind socket to IP address and port
// Accept incoming connections
```

Tutorial

C++