Python: Apache Tomcat Cookbook (Nutshell Examples)

Luka Benzia

Seminar BIS

SS 2021

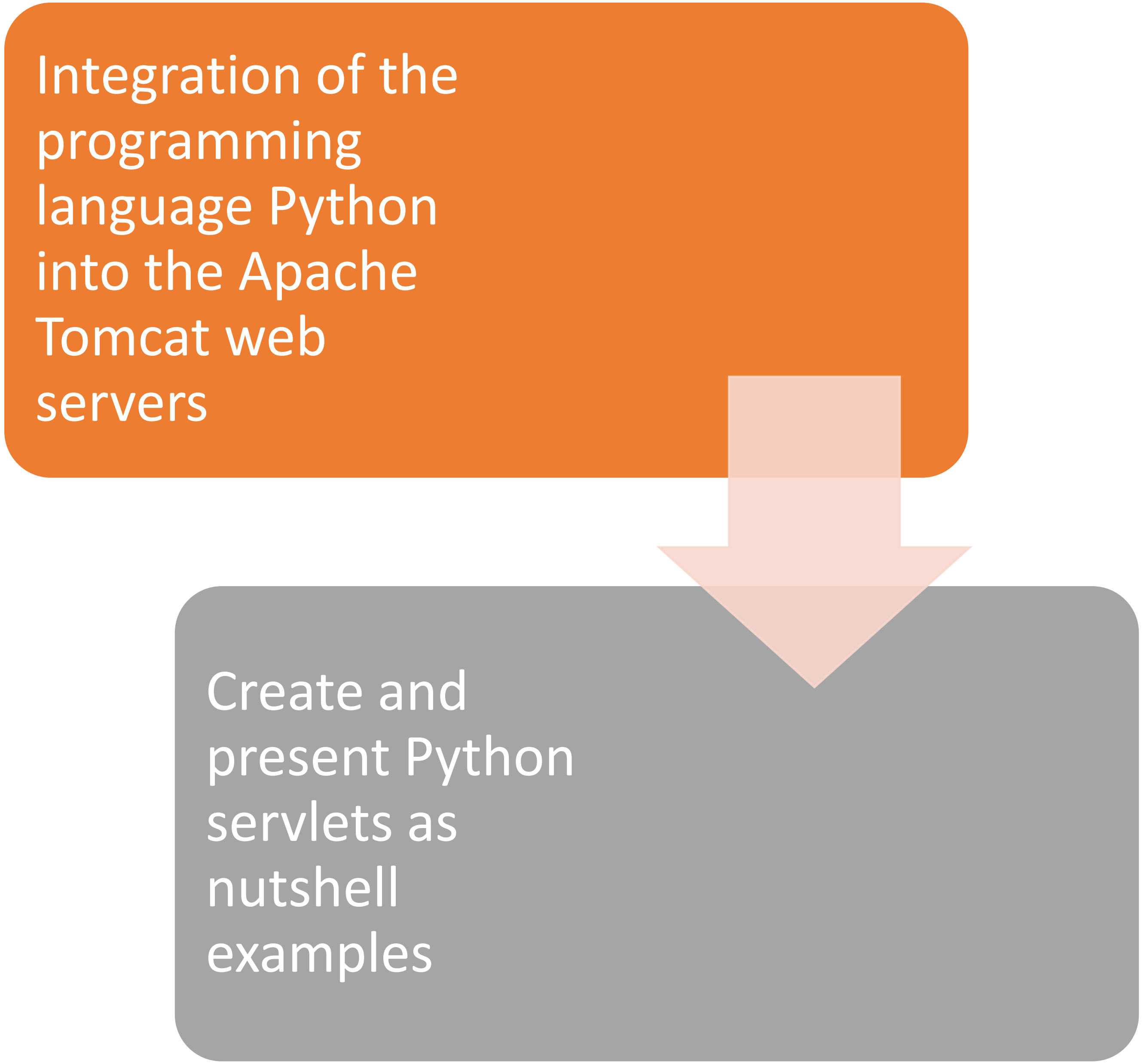
Content

1. Introduction
2. Theoretical background
    1. Apache Tomcat
    2. Python
    3. Jython
3. Nutshell Examples
    1. Hello World
    2. Request Information
    3. Request Header
    4. Request Parameter
    5. Cookies
4. Conclusion
    1. Summary
    2. Outlook

1. Introduction

- 1. Goal of the paper
- 2. table of contents of the Seminar Paper

1. Goal of the Paper



Integration of the programming language Python into the Apache Tomcat web servers

Create and present Python servlets as nutshell examples

2. Table of contents of the seminar paper

- **1. Introduction**
  - 1.1 Objectives
  - 1.2 Methodology
  - 1.3 Structure

- **2. Background**
  - 2.1 Apache Tomcat
  - 2.2 Python
  - 2.3 Jython

- **3. Nutshell Examples**
  - 3.1 Hello World
  - 3.2 Request Information
  - 3.3 Request Header
  - 3.4 Request Parameter
  - 3.5 Cookies

- **5. Conclusion**
  - 5.1 Summary
  - 5.2 Outlook

- **Appendix**
  - Installation Guide

2. Theoretical Background

- 1. Apache Tomcat
- 2. Python
- 3. Jython

1. Apache Tomcat History

- Startet as a project by James Duncan Davidson at Sun Microsystems
- Tomcat codes base was transferred to the Apache Software Foundation in 1999
- Was continued as an Open-Source project
- Got his own management structure in 2005

1. Apache Tomcat Description

- Web Application server designed to execute Java servlets
- Open-Source Software
  - Free for everyone to use
  - Gives user complete freedom over his web servers
- Lightweight Application
  - Offers most basic functions for a server
  - Almost No loading times
  - Redeploy time much faster than other applications of this type
- Stability
  - Runs independently of the Apache installation
  - Even a major error in Tomcat does not cause a crash
- Very good Documentation

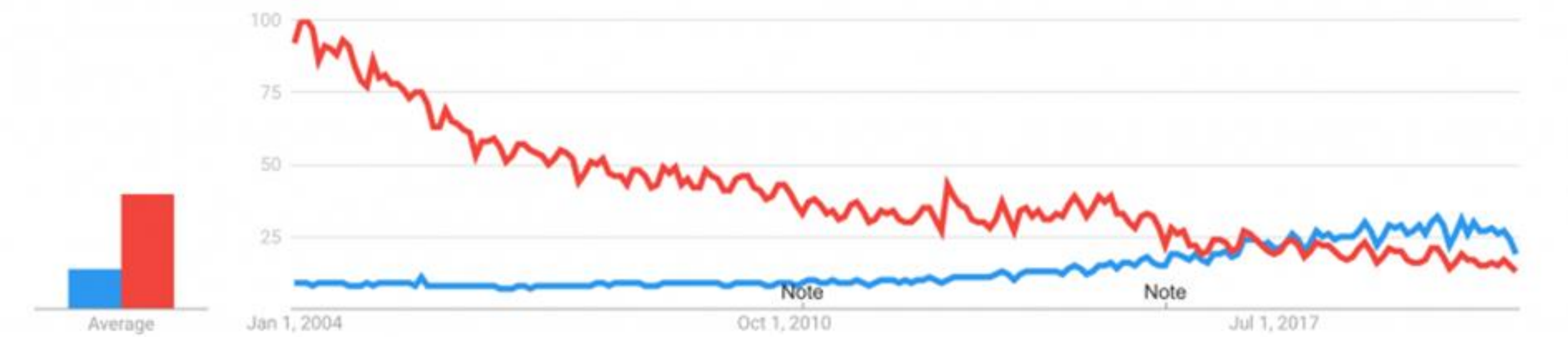
2. Python History

- Work started at the end of 1980 by Guido Van Rossum
- Based on ABC programming language
  - Syntax
  - Best features
  - Issues fixed
- Finally released in 1991

2. Python Description

- Interpreted
- object-oriented
- high-level programming language with dynamic semantics.
- Simple language and easy to learn the syntax compared to other languages
- Can be freely ditributed

Java vs. Python



3. Jython History

- Created in 1997 by Jim Hugunin
- originally made to replace C with Java for performance-intensive code accessed by Python programs.
- Moved to SourceForge in 2000
- Awarded a grant by the Python Software Foundation in 2005

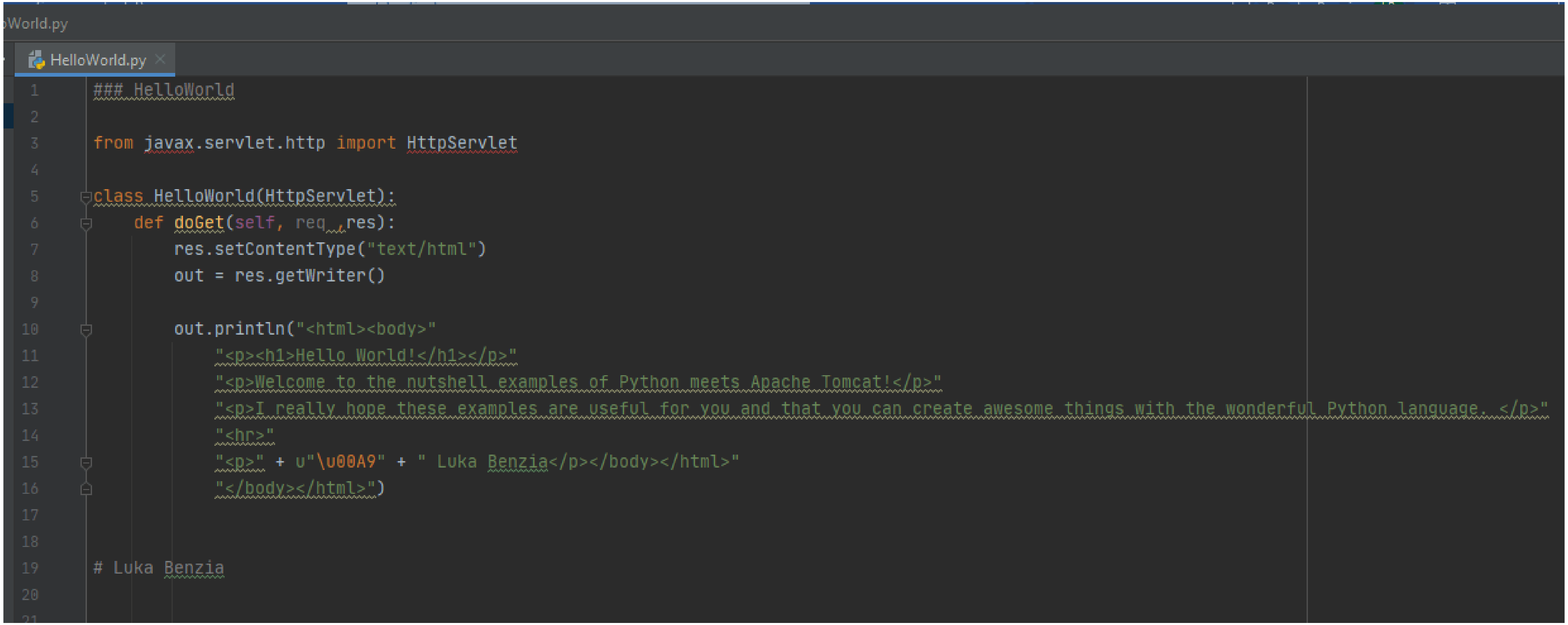
3. Jython Description

- Java implementation of Python
- Combines expressive power with clarity
- Freely available
- Needed to create Python servlets

3. Nutshell Examples

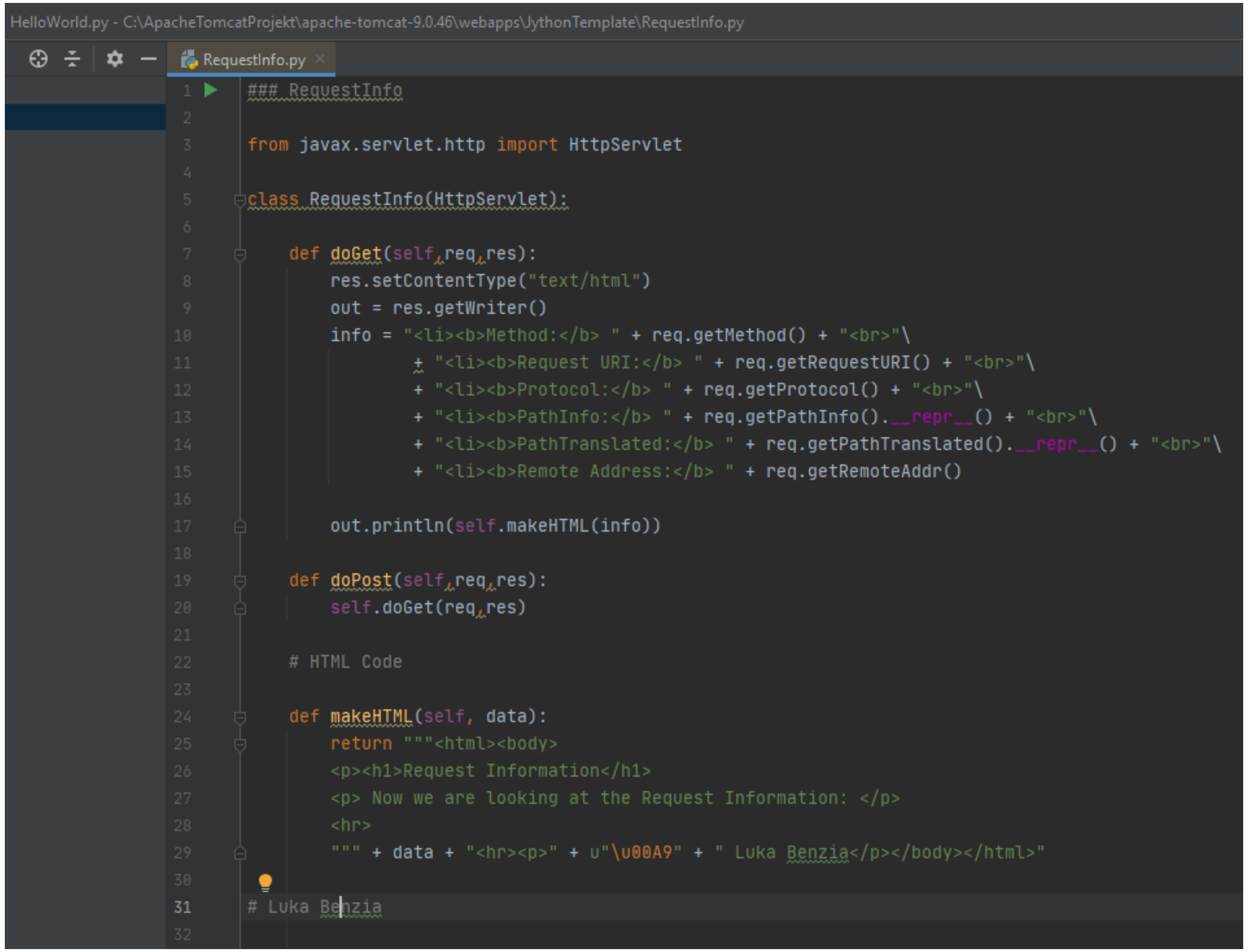
- 1. Hello World
- 2. Request Information
- 3. Request Header
- 4. Request Parameter
- 5. Cookies

1. Hello World



```
### HelloWorld

from javax.servlet.http import HttpServlet

class HelloWorld(HttpServlet):
    def doGet(self, req ,res):
        res.setContentType("text/html")
        out = res.getWriter()

        out.println("<html><body>"
            "<p><h1>Hello World!</h1></p>"
            "<p>Welcome to the nutshell examples of Python meets Apache Tomcat!</p>"
            "<p>I really hope these examples are useful for you and that you can create awesome things with the wonderful Python language. </p>"
            "<hr>"
            "<p>" + u"\u00A9" + " Luka Benzia</p></body></html>"
            "</body></html>")


# Luka Benzia
```

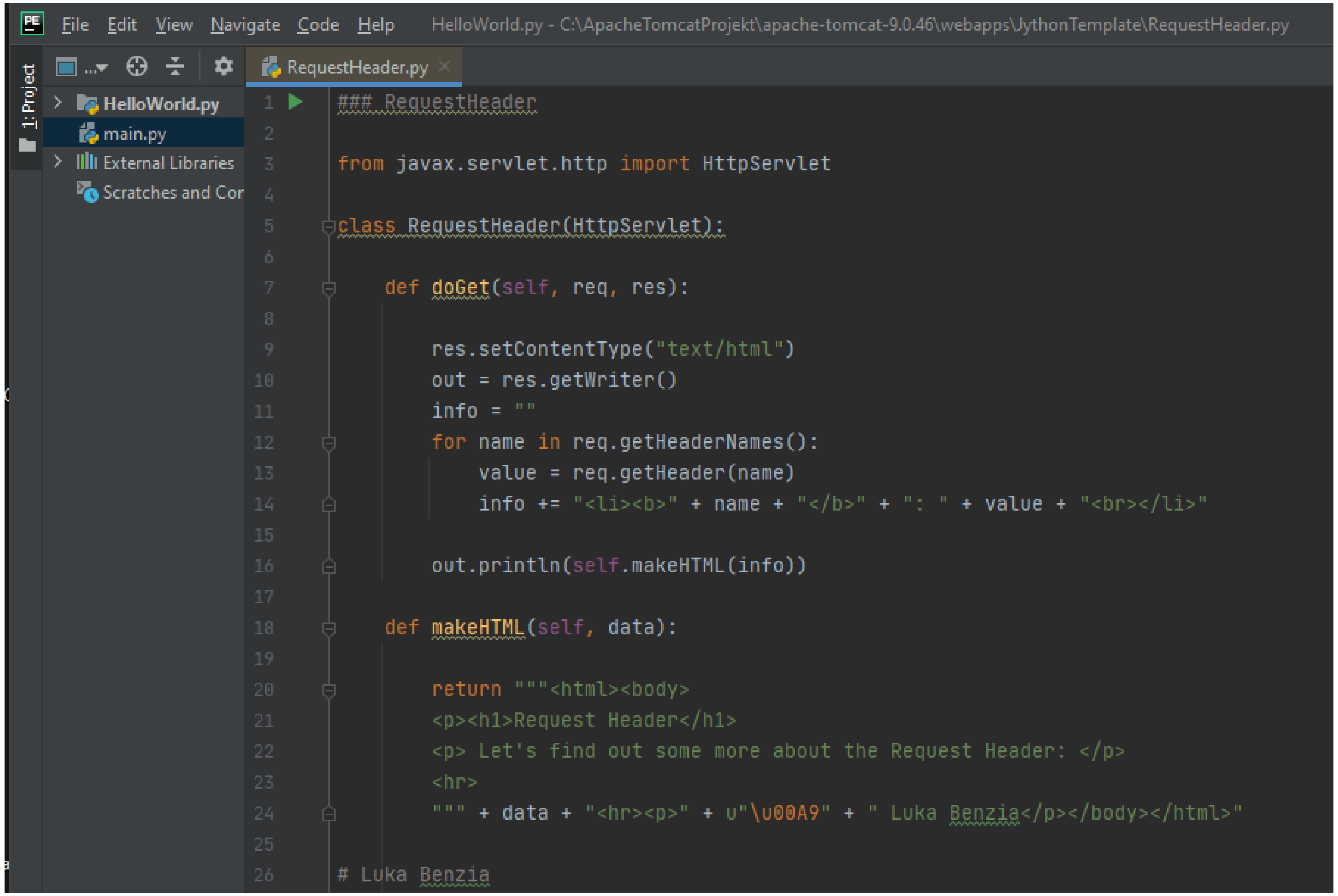
2. Request Info



HelloWorld.py - C:\ApacheTomcatProjekt\apache-tomcat-9.0.46\webapps\JythonTemplate\RequestInfo.py
RequestInfo.py

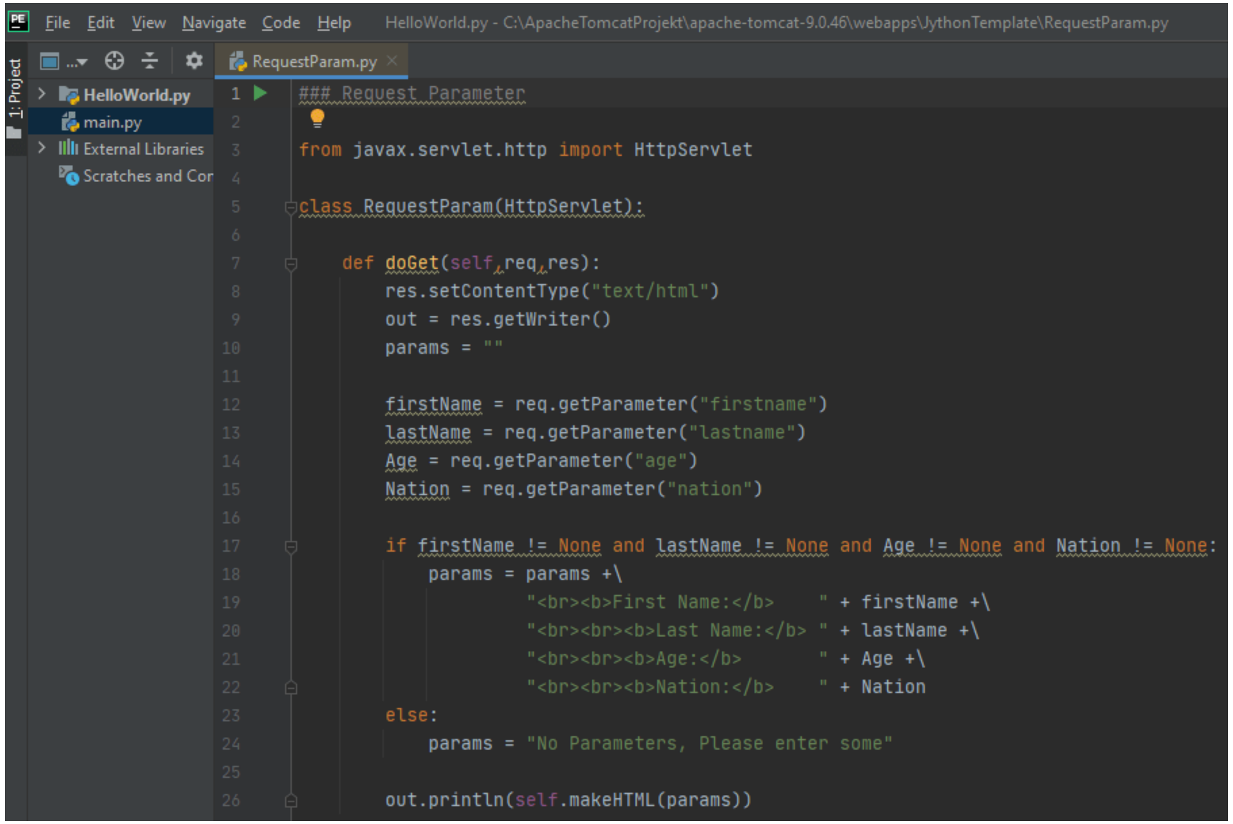
```python
### RequestInfo

from javax.servlet.http import HttpServlet

class RequestInfo(HttpServlet):

    def doGet(self,req,res):
        res.setContentType("text/html")
        out = res.getWriter()
        info = "<li><b>Method:</b> " + req.getMethod() + "<br>"\
                + "<li><b>Request URI:</b> " + req.getRequestURI() + "<br>"\
                + "<li><b>Protocol:</b> " + req.getProtocol() + "<br>"\
                + "<li><b>PathInfo:</b> " + req.getPathInfo().__repr__() + "<br>"\
                + "<li><b>PathTranslated:</b> " + req.getPathTranslated().__repr__() + "<br>"\
                + "<li><b>Remote Address:</b> " + req.getRemoteAddr()

        out.println(self.makeHTML(info))

    def doPost(self,req,res):
        self.doGet(req,res)

    # HTML Code

    def makeHTML(self, data):
        return """<html><body>
        <p><h1>Request Information</h1>
        <p> Now we are looking at the Request Information: </p>
        <hr>
        """ + data + "<hr><p>" + u"\u00A9" + " Luka Benzia</p></body></html>"

# Luka Benzia

```

3. Request Header

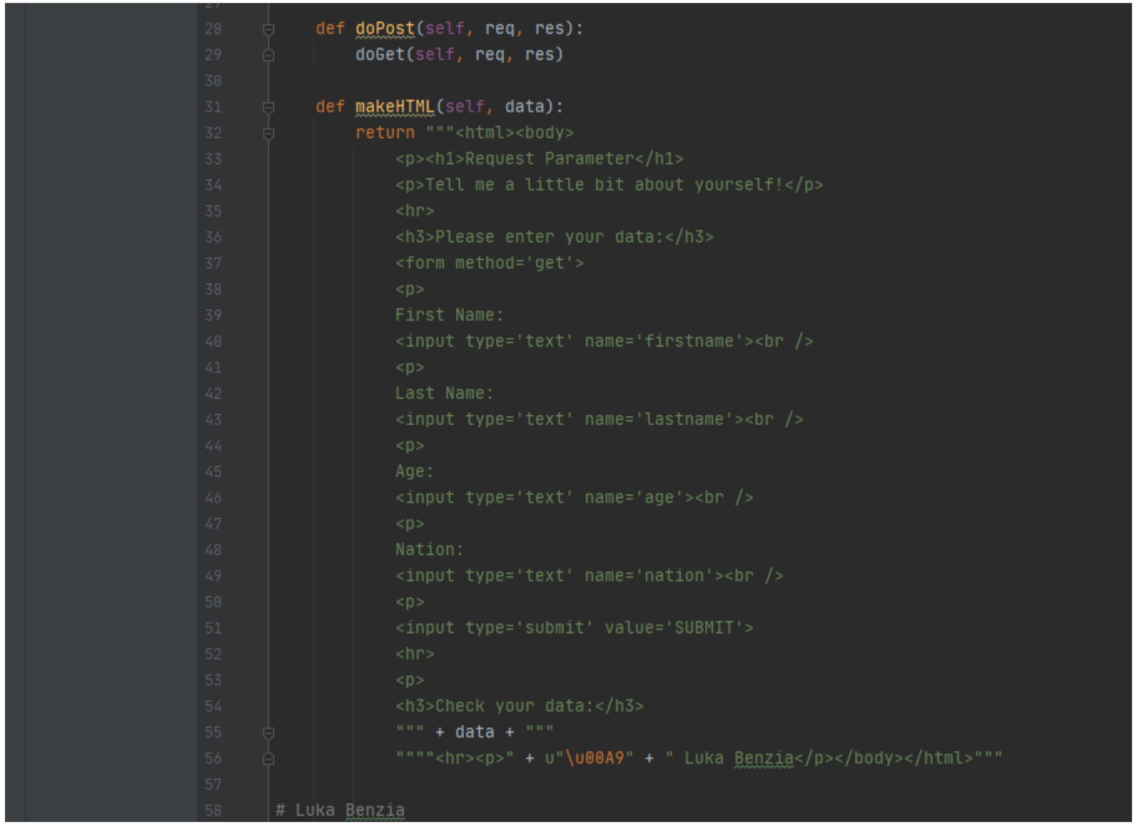


```python
### RequestHeader

from javax.servlet.http import HttpServlet

class RequestHeader(HttpServlet):

    def doGet(self, req, res):

        res.setContentType("text/html")
        out = res.getWriter()
        info = ""
        for name in req.getHeaderNames():
            value = req.getHeader(name)
            info += "<li><b>" + name + "</b>" + ": " + value + "<br></li>"

        out.println(self.makeHTML(info))

    def makeHTML(self, data):

        return """<html><body>
        <p><h1>Request Header</h1>
        <p> Let's find out some more about the Request Header: </p>
        <hr>
        """ + data + "<hr><p>" + u"\u00A9" + " Luka Benzia</p></body></html>"

# Luka Benzia
```

4. Request Parameter

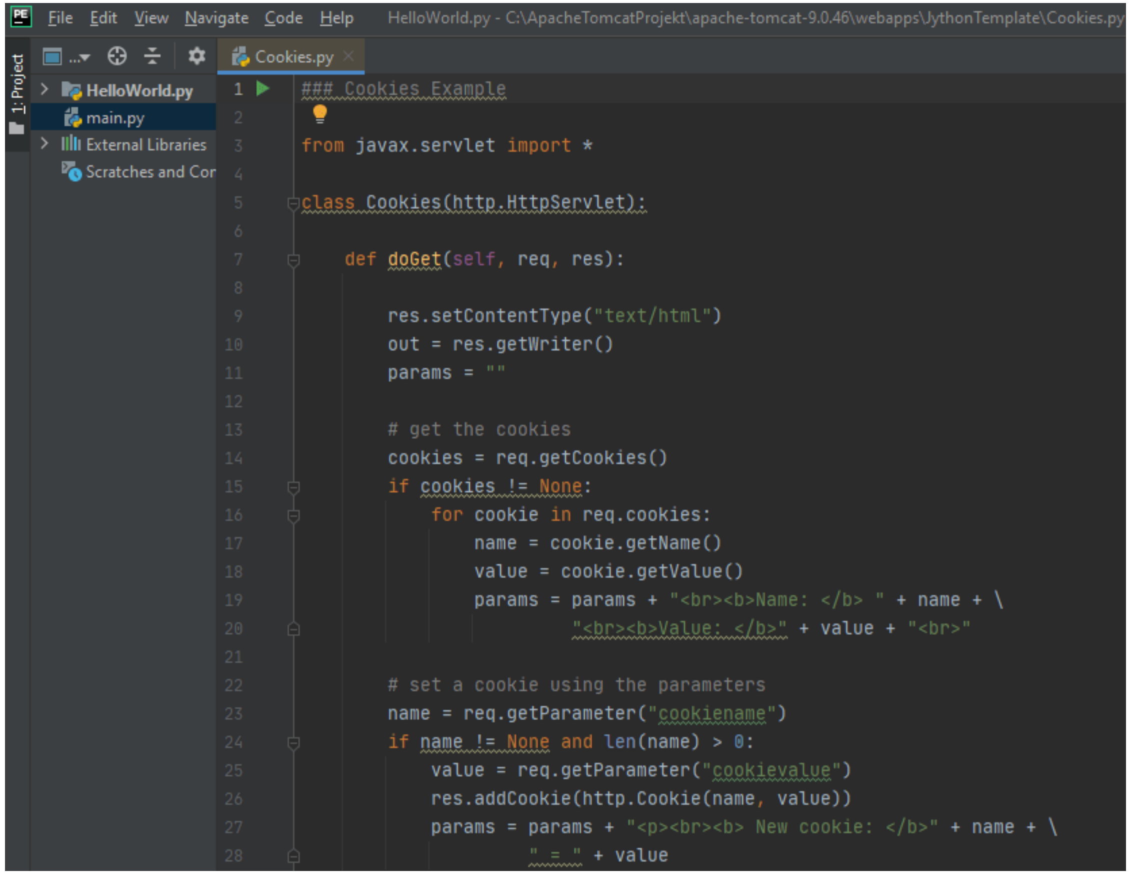


```python
### Request Parameter

from javax.servlet.http import HttpServlet

class RequestParam(HttpServlet):

    def doGet(self,req,res):
        res.setContentType("text/html")
        out = res.getWriter()
        params = ""

        firstName = req.getParameter("firstname")
        lastName = req.getParameter("lastname")
        Age = req.getParameter("age")
        Nation = req.getParameter("nation")

        if firstName != None and lastName != None and Age != None and Nation != None:
            params = params +\
                     "<br><b>First Name:</b>     " + firstName +\
                     "<br><br><b>Last Name:</b> " + lastName +\
                     "<br><br><b>Age:</b>       " + Age +\
                     "<br><br><b>Nation:</b>    " + Nation
        else:
            params = "No Parameters, Please enter some"

        out.println(self.makeHTML(params))
```

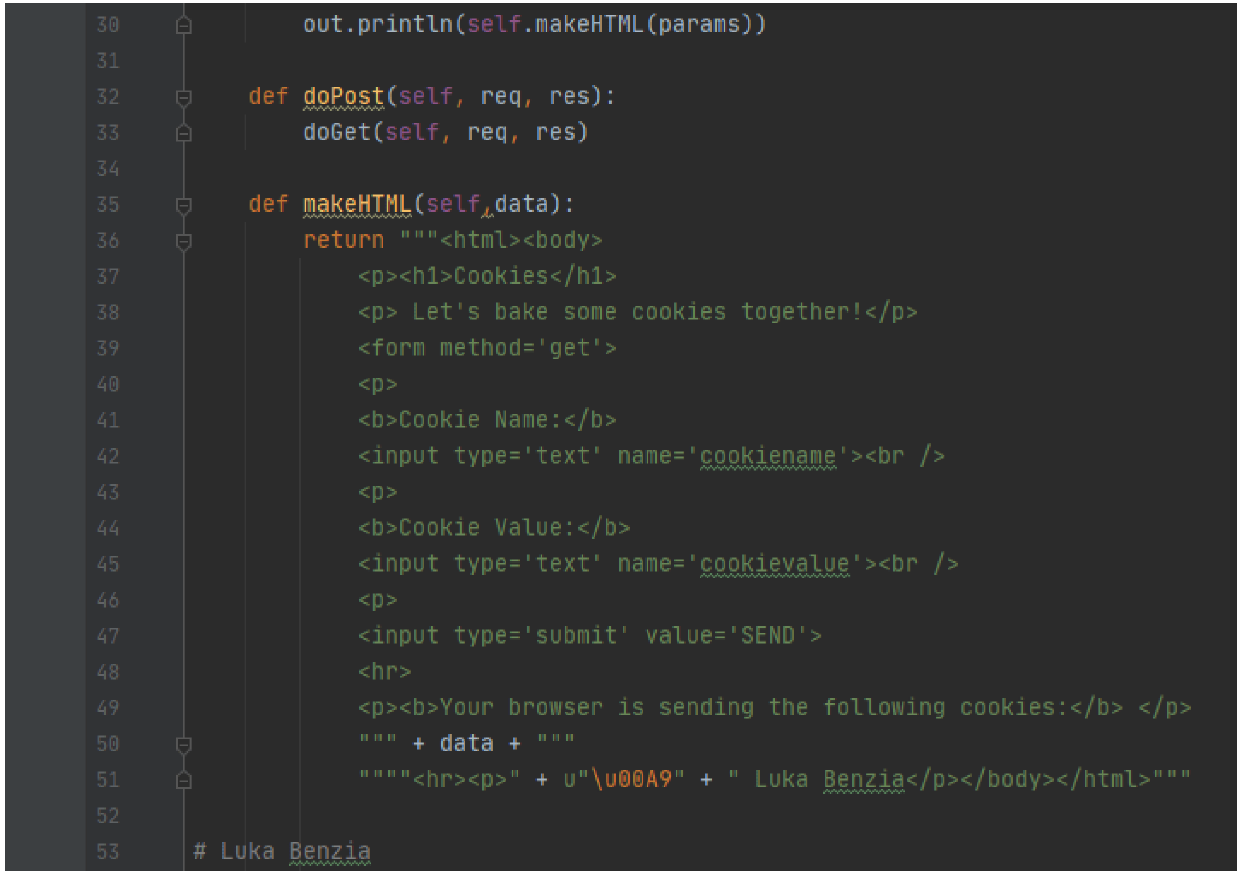


```python
    def doPost(self, req, res):
        doGet(self, req, res)

    def makeHTML(self, data):
        return """<html><body>
            <p><h1>Request Parameter</h1>
            <p>Tell me a little bit about yourself!</p>
            <hr>
            <h3>Please enter your data:</h3>
            <form method='get'>
            <p>
            First Name:
            <input type='text' name='firstname'><br />
            <p>
            Last Name:
            <input type='text' name='lastname'><br />
            <p>
            Age:
            <input type='text' name='age'><br />
            <p>
            Nation:
            <input type='text' name='nation'><br />
            <p>
            <input type='submit' value='SUBMIT'>
            <hr>
            <p>
            <h3>Check your data:</h3>
            """ + data + """
            """"<hr><p>" + u"\u00A9" + " Luka Benzia</p></body></html>"""

# Luka Benzia
```

5. Cookies


File Edit View Navigate Code Help
HelloWorld.py - C:\ApacheTomcatProjekt\apache-tomcat-9.0.46\webapps\JythonTemplate\Cookies.py
1: Project
Cookies.py
HelloWorld.py
main.py
External Libraries
Scratches and Cor

```
### Cookies Example

from javax.servlet import *

class Cookies(http.HttpServlet):

    def doGet(self, req, res):

        res.setContentType("text/html")
        out = res.getWriter()
        params = ""

        # get the cookies
        cookies = req.getCookies()
        if cookies != None:
            for cookie in req.cookies:
                name = cookie.getName()
                value = cookie.getValue()
                params = params + "<br><b>Name: </b> " + name + \
                         "<br><b>Value: </b>" + value + "<br>"

        # set a cookie using the parameters
        name = req.getParameter("cookiename")
        if name != None and len(name) > 0:
            value = req.getParameter("cookievalue")
            res.addCookie(http.Cookie(name, value))
            params = params + "<p><br><b> New cookie: </b>" + name + \
                     " = " + value
```



```
        out.println(self.makeHTML(params))

    def doPost(self, req, res):
        doGet(self, req, res)

    def makeHTML(self,data):
        return """<html><body>
            <p><h1>Cookies</h1>
            <p> Let's bake some cookies together!</p>
            <form method='get'>
            <p>
            <b>Cookie Name:</b>
            <input type='text' name='cookiename'><br />
            <p>
            <b>Cookie Value:</b>
            <input type='text' name='cookievalue'><br />
            <p>
            <input type='submit' value='SEND'>
            <hr>
            <p><b>Your browser is sending the following cookies:</b> </p>
            """ + data + """
            """"<hr><p>" + u"\u00A9" + " Luka Benzia</p></body></html>"""

# Luka Benzia
```

4. Conclusion

- 1. Summary
- 2. Outlook

1. Summary

1

Apache Tomcat

- easy to install
- well documented
- offers numerous features

2

Python

- simplifies the creation of Servlets, as less typing is required
- more forgiving

2. Outlook

1

Implementing other programming languages

2

Creation of more complex Nutshell Examples