University of Gdańsk

Subject card

Subject name and code	Script Languages, PG_00198459						
Field of study	Informatics						
Date of commencement of studies	October 2026		Academic year of realisation of subject	2026/2027			
Education level	Bachelor's studies		Subject group	Obligatory subject group in the field of study			
Mode of study	part-time studies		Mode of delivery	at the university			
Year of study	1		Language of instruction	Polish			
Semester of study	2		ECTS credits	3.0			
Learning profile	academic		Assessment form	credit			
Conducting unit							
Name and surname of lecturer (lecturers)	Subject supervisor	dr Piotr Arłukowicz					
	Teachers						

Lesson types

Lesson type	Lecture	Tutorial	Laboratory	Project	Seminar	SUM
Number of study hours	10.0	0.0	10.0	0.0	0.0	20

E-learning hours included: 0.0

Learning activity and number of study hours

Learning activity	Participation in didactic classes included in study plan	Participation in consultation hours	Self-study	SUM
Number of study hours	20	0.0	55.0	75

Subject objectives

To familiarize students with the scripting programming paradigm and the practical use of scripting languages for task automation, data and text processing, and rapid prototyping.

Learning outcomes

Course outcome	Subject outcome	Method of verification
[INFOL3_W05] knows and understands advanced concepts in the field of various programming paradigms and programming languages; as well as object-oriented design and programming methods and patterns	Knows and understands the characteristics of scripting languages and their place among paradigms — dynamic typing, interpretation, and procedural/OO/ functional programming in a scripting language.	[SW4] test/exam - oral or written [SW2] presentation/project/paper/ report
[INFOL3_W04] knows and understands advanced concepts in the field of software engineering, software specifications, validation and verification, and tools supporting the software development process	Knows the scripting-language ecosystem tools supporting development — package and runtime-environment management, testing, debugging and version control.	[SW4] test/exam - oral or written [SW2] presentation/project/paper/ report
[INFOL3_U06] is able to select and apply appropriate IT methods and tools to solve complex problems, is able to use acquired knowledge by appropriately selecting sources and information derived from them, and perform evaluation, critical analysis, and synthesis of this information	Is able to select a scripting language and appropriate libraries for a task and write a script automating processing of data, files and system tasks, using documentation and other sources.	[SU2] presentation/project/paper/ report [SU4] test/exam - oral or written

<table>
<tr><td>Subject contents</td><td>1. Intro to scripting languages — interpretation, dynamic typing, applications, vs compiled languages.
2. Fundamentals — data types, variables, control statements, functions.
3. Built-in data structures — lists, tuples, dictionaries, sets; operations and iteration.
4. Text processing and regular expressions.
5. OOP and elements of functional programming in a scripting language.
6. Modules, packages, code organisation; environment and package management.
7. File/system operations; data-exchange formats (JSON, CSV, XML).
8. Shell interaction and task automation.
9. Error/exception handling, debugging, basics of testing.
10. Version control and good practices.</td></tr>
<tr><td>Prerequisites and co-requisites</td><td></td></tr>
<tr><td>Assessment methods and criteria</td><td><table>
<tr><th>Subject passing criteria</th><th>Passing threshold</th><th>Percentage of the final grade</th></tr>
<tr><td>project/lab assignements</td><td>51.0%</td><td>60.0%</td></tr>
<tr><td>test</td><td>51.0%</td><td>40.0%</td></tr>
</table></td></tr>
<tr><td>Recommended reading</td><td><table>
<tr><td>Basic literature</td><td>• M. Lutz, Learning Python (Polish ed., Helion).
• M. Summerfield, Programming in Python 3 (Polish ed., Helion).
• Python language documentation: https://docs.python.org/</td></tr>
<tr><td>Supplementary literature</td><td>• A. Sweigart, Automate the Boring Stuff with Python (Polish ed., Helion).
• L. Ramalho, Fluent Python (Polish ed., Helion).
• System shell documentation (Bash): https://www.gnu.org/software/bash/manual/</td></tr>
<tr><td>eResources addresses</td><td></td></tr>
</table></td></tr>
<tr><td>Example issues/ example questions/ tasks being completed</td><td>• Process text files with regular expressions.
• Implement a class; use functional elements (map/filter/reduce, lambdas).
• Read/transform JSON/CSV data and write the result.
• Automate a system task (file ops, shell commands).
• Organise code into modules/package; configure a runtime environment.
• Handle exceptions; write a simple unit test.</td></tr>
<tr><td>Work placement</td><td>Not applicable</td></tr>
</table>

Document generated electronically. Does not require a seal or signature.