



OPIS ZAJĘĆ/PRZEDMIOTU (SYLABUS)

I. Informacje ogólne

1. Nazwa zajęć/przedmiotu – **IT Skills for Linguists 2**
2. Kod zajęć/przedmiotu – 15-ITL2-EL-12
3. Rodzaj zajęć/przedmiotu (obowiązkowy lub fakultatywny) – obowiązkowy
4. Kierunek studiów – English Linguistics: Theories, Interfaces, Technologies
5. Poziom studiów (I lub II stopień, jednolite studia magisterskie) – poziom I
6. Profil studiów (ogólnoakademicki / praktyczny) – ogólnoakademicki
7. Rok studiów (jeśli obowiązuje) – 2 rok
8. Rodzaje zajęć i liczba godzin (np.: 15 h W, 30 h ĆW) – 60 h ćw.
9. Liczba punktów ECTS – 3
10. Imię, nazwisko, tytuł / stopień naukowy, adres e-mail wykładowcy (wykładowców*) / prowadzących zajęcia – dr Grzegorz Krynicki
11. Język wykładowy – angielski
12. Moduł zajęć / przedmiotu prowadzony zdalnie (e-learning) (tak [częściowo/w całości] / nie) – tak, w całości

*proszę podkreślić koordynatora przedmiotu

II. Informacje szczegółowe

1. Cele zajęć/przedmiotu

- Ogólne zapoznanie Studenta z zagadnieniami dotyczącymi programowania skryptowego
- Pierwsze wprowadzenie do wybranych właściwości języka programowania Python, takich jak funkcje, zmienne, przepływ sterowania czy struktury danych
- Rozwinięcie u Studenta umiejętności posługiwania się językiem programowania Python w celu automatyzacji powtarzalnych czynności związanych z przetwarzaniem tekstu
- Zaznajomienie Studenta z wybranymi bibliotekami dostępnymi dla języka programowania Python, służącymi do przeprowadzania analizy danych językowych.

2. Wymagania wstępne w zakresie wiedzy, umiejętności oraz kompetencji społecznych (jeśli obowiązują)

- Umiejętność wykorzystania komputera, instalacja oprogramowania, edytor tekstu
- Znajomość języka angielskiego na poziomie B1

3. Efekty uczenia się (EU) dla zajęć i odniesienie do efektów uczenia się dla kierunku studiów:

Symbol EU dla modułu zajęć/przedmiotu	Po zakończeniu modułu i potwierdzeniu osiągnięcia EK student /ka:	Symbole EK dla kierunku studiów
EU_01	Potrafi samodzielnie przygotować stanowisko i narzędzia do pracy z językiem programowania Python.	K_W09, K_U05
EU_02	Rozumie i potrafi posługiwać się składnią języka programowania Python.	K_W04, K_U05
EU_03	Potrafi zastosować odpowiednie typy i struktury danych oraz składnię w celu uzyskania zamierzonej logiki działania programu.	K_U09
EU_04	Potrafi utworzyć skrypt w języku programowania Python umożliwiający jego wielokrotne zastosowanie.	K_U04, K_U05, K_U09
EU_05	Umie odczytywać, zapisywać oraz konwertować dane przy pomocy języka programowania Python.	K_U09
EU_06	Umie korzystać z wyrażeń regularnych do konwersji danych tekstowych.	K_W06, K_U09



EU_07	Jest w stanie samodzielnie przeprowadzić podstawową analizę danych językowych.	K_W01, K_W06, K_U03, K_U09
EU_08	Potrafi współpracować z innymi osobami nad opracowaniem kodu programistycznego.	K_U01, K_U10, K_U12
EU_09	Umie wykorzystać biblioteki programistyczne opracowane przez osoby trzecie w celu realizacji zadań programistycznych.	K_W04, K_W08, K_U02, K_U09
EU_10	Potrafi samodzielnie odnajdywać informacje i materiały o tematyce programistycznej.	K_U02, K_K01, K_K03
EU_11	Uświadamia sobie potrzebę dalszego doskonalenia swoich umiejętności oraz wie, jak to robić.	K_U03, K_K01, K_K03,
EU_12	Uświadamia sobie pola zastosowań poznanych technik w kontekstach komercyjnych.	K_K05

4. Treści programowe zapewniające uzyskanie efektów uczenia się (EU) z odniesieniem do odpowiednich efektów uczenia się (EU) dla zajęć/przedmiotu

Treści programowe dla zajęć/przedmiotu	Symbol/symbole EU dla zajęć / przedmiotu
Instalacja i konfiguracja środowiska programistycznego	EU_01, EU_10
Korzystanie z narzędzi programistycznych	EU_01, EU_04, EU_08, EU_10
Składnia języka programowania Python	EU_02–EU_04, EU_10
Typy i struktury danych	EU_03–EU_05, EU_10
Funkcje	EU_02, EU_04, EU_07
Przeływ sterowania w programowaniu	EU_02, EU_04
Przetwarzanie i analiza plików tekstowych	EU_05, EU_07
Konwersja danych	EU_05, EU_06, EU_12
Wyszukiwanie i podmiana tekstu przy pomocy wyrażeń regularnych	EU_06
Programowanie skryptowe	EU_03, EU_04, EU_08, EU_12
Przetwarzanie danych językowych przy pomocy bibliotek programistycznych	EU_08, EU_09, EU_10, EU_11

5. Zalecana literatura:

- **Hammond, Michael. 2020. Python for Linguists. Cambridge University Press.**
- Hetland, Magnus Lie. 2017. Beginning Python: From Novice to Professional. (3rd edition.) Berkeley: Apress.
- Sweigart, Al. 2015. Automate The Boring Stuff With Python. (<https://automatetheboringstuff.com/>)
- McKinney, Wes. 2017. Python For Data Analysis. (2nd edition.) Sebastopol: O'Reilly Media.
- Langtangen, Hans Petter. 2004. Python scripting for computational science. Berlin: Springer.
- Lutz, Mark. 2013. Learning Python. (5th edition.) Sebastopol: O'Reilly Media.

III. Informacje dodatkowe

1. Metody i formy prowadzenia zajęć umożliwiające osiągnięcie założonych EU (proszę wskazać z proponowanych metod właściwe dla opisywanych zajęć lub/i zaproponować inne)



Metody i formy prowadzenia zajęć	
1. Wykład z prezentacją multimedialną wybranych zagadnień	✓
2. Wykład konwersatoryjny	✓
3. Wykład problemowy	
4. Dyskusja	✓
5. Praca z tekstem	
6. Metoda analizy przypadków	✓
7. Uczenie problemowe (Problem-based learning)	✓
8. Gra dydaktyczna/symulacyjna	
9. Rozwiązywanie zadań (np.: obliczeniowych, artystycznych, praktycznych)	✓
10. Metoda ćwiczeniowa	✓
11. Metoda laboratoryjna	
12. Metoda badawcza (dociekania naukowego)	
13. Metoda warsztatowa	
14. Metoda projektu	✓
15. Pokaz i obserwacja	
16. Demonstracje dźwiękowe i/lub video	✓
17. Metody aktywizujące (np.: „burza mózgów”, technika analizy SWOT, technika drzewka decyzyjnego, metoda „kuli śniegowej”, konstruowanie „map myśli”)	
18. Praca w grupach	✓
19. Zadania na platformie e-learningowej Moodle, https://www.onlinegdb.com , https://codeboard.io , https://repl.it	✓

2. Sposoby oceniania stopnia osiągnięcia EU (proszę wskazać z proponowanych sposobów właściwe dla danego EU lub/i zaproponować inne)

	EU_01	EU_02	EU_03	EU_04	EU_05	EU_06	EU_07	EU_08	EU_09	EU_10	EU_11	EU_12	EU_13	EU_14
Test		✓	✓	✓	✓	✓	✓		✓					
Projekt	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Raport		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Zadanie domowe wykonania program komputerowego	✓	✓	✓	✓	✓	✓	✓	✓		✓				
Zadanie klasowe wykonania programu komputerowego	✓	✓	✓	✓	✓	✓	✓		✓					

3. Nakład pracy studenta i punkty ECTS

Forma aktywności	Średnia liczba godzin na zrealizowanie aktywności
Godziny zajęć (wg planu studiów) z nauczycielem	60
☒ ☑ Przygotowanie do zajęć	10



	Czytanie wskazanej literatury	10
	Przygotowanie raportu i prezentacji	20
	Przygotowanie projektu	20
	Przygotowanie pracy semestralnej	
	Przygotowanie do egzaminu / zaliczenia	
	...	
SUMA GODZIN		120
LICZBA PUNKTÓW ECTS DLA MODUŁU ZAJĘĆ/PRZEDMIOTU		3

* proszę wskazać z proponowanych przykładów pracy własnej studenta właściwe dla opisywanego modułu lub/i zaproponować inne

4. Kryteria oceniania wg skali stosowanej w UAM:

- Ocena będzie średnią ważoną następujących składowych
 - Przygotowanie do zaj. (10%),
 - Zadania klasowe (10%),
 - Zadania domowe (20%),
 - Duże testy (30%),
 - Projekt (30%)
- Średnia posłuży do wyznaczenia oceny wg następującej skali
 - bardzo dobry (bdb; 5,0): znakomita wiedza i umiejętności
 - dobry plus (+db; 4,5): bardzo dobra wiedza i umiejętności
 - dobry (db; 4,0): dobra wiedza i umiejętności
 - dostateczny plus (+dst; 3,5): dobra wiedza i umiejętności, ale z niedociągnięciami
 - dostateczny (dst; 3,0): zadowalająca wiedza i umiejętności
 - niedostateczny (ndst; 2,0): niezadowalająca wiedza i umiejętności



SYLLABUS – A COURSE DESCRIPTION

I. General information

1. Course name: **Elective 3: Natural Language Processing / Linguistic epistemology / Lexicography**
2. Course code: 15-EL3-EL-11
3. Course type (compulsory or optional): compulsory
4. Study programme name: English Linguistics: Theories, Interfaces, Technologies
5. Cycle of studies: 1st cycle
6. Educational profile (general academic profile or practical profile): academic
7. Year of studies (if relevant): 2
8. Type of classes and number of contact hours: tutorial 60 h
9. Number of ECTS credits: 3
10. Name, surname, academic degree/title, and e-mail address of the module lecturer (lecturers*) / teachers: dr Grzegorz Krynicki
11. Language of instruction: English
12. Online learning (yes (partially / entirely) / no) – yes, entirely

*please underline module coordinator's name

II. Detailed information

1. Course aim (aims)
 - Introduce students to Python programming language: functions, variables, control flow or data structures
 - Use Python programs to help you manipulate and analyse basic data incl. simple language data
2. Pre-requisites in terms of knowledge, skills and social competences (if relevant)
 - Using file managers, installing software, using text editors, regular expression (*IT Skills for Linguists 1*)
 - English at B1
3. Course learning outcomes (EU) in terms of knowledge, skills and social competences and their reference to study programme learning outcome

Course learning outcome symbol (EU)	On successful completion of the course and validation of its learning outcomes, a student:	Reference to study programme learning outcomes
EU_01	Student can independently prepare a stand and tools to work with the Python	K_W09, K_U05
EU_02	Student understands and is able to use the syntax of the Python programming	K_W04, K_U05
EU_03	Student can use appropriate data types and structures and syntax to achieve	K_U09
EU_04	Student can create a script in the Python programming language for	K_U04, K_U05, K_U09
EU_05	Student can read, write and convert data using the Python programming	K_U09



EU_06	Student is able to use regular expressions to convert text data	K_W06, K_U09
EU_07	Student is able to perform basic analysis of language data on its own	K_W01, K_W06, K_U03, K_U09
EU_08	Student is able to cooperate with other people on the development of	K_U01, K_U10, K_U12
EU_09	Student is able to use programming libraries developed by third parties in	K_W04, K_W08, K_U02, K_U09
EU_10	Student can independently find information on programming topics	K_U02, K_K01, K_K03
EU_11	Student realizes the need to further improve skills and knows how to do it	K_U03, K_K01, K_K03,
EU_12	Student is aware of the fields of application of the learned techniques in	K_K05

4. Learning content with reference to course learning outcomes (EU)

Course learning content	Course learning outcome symbol(s) (EU)
Installation and configuration of the development environment	EU_01, EU_10
Use of programming tools	EU_01, EU_04, EU_08, EU_10
Python programming language syntax	EU_02–EU_04, EU_10
Data types and structures	EU_03–EU_05, EU_10
Functions	EU_02, EU_04, EU_07
Control flow in programming	EU_02, EU_04
Processing and analysis of text files	EU_05, EU_07
Data conversion	EU_05, EU_06, EU_12
Searching for and exchanging text with regular expressions	EU_06
Scriptural programming	EU_03, EU_04, EU_08, EU_12
Processing of language data using programming libraries	EU_08, EU_09, EU_10, EU_11

5. Reading list

- **Hammond, Michael. 2020. Python for Linguists. Cambridge University Press.**
- Hetland, Magnus Lie. 2017. Beginning Python: From Novice to Professional. (3rd edition.) Berkeley: Apress.
- Sweigart, Al. 2015. Automate The Boring Stuff With Python. (<https://automatetheboringstuff.com/>)
- McKinney, Wes. 2017. Python For Data Analysis. (2nd edition.) Sebastopol: O'Reilly Media.
- Langtangen, Hans Petter. 2004. Python scripting for computational science. Berlin: Springer.
- Lutz, Mark. 2013. Learning Python. (5th edition.) Sebastopol: O'Reilly Media.



III. Additional information

- Teaching and learning methods and activities to enable students to achieve the intended course learning outcomes (please indicate the appropriate methods and activities with a tick and/or suggest other methods.)

Teaching methods and techniques	
1. Lecture with multimedia presentation of selected issues	✓
2. Interactive lecture	✓
3. Lecture with a focus on problem-solving	
4. Discussion	✓
5. Working with text	
6. Case study method	✓
7. Problem-based learning	✓
8. Teaching/simulation game	
9. Solving tasks (e.g.: computational, artistic, practical)	✓
10. Exercise method	✓
11. Laboratory method	
12. Research method (scientific inquiry)	
13. Workshop method	
14. Project method	✓
15. Showdown and observation	
16. Audio and/or video demonstrations	✓
17. Activation methods (e.g.: "brainstorming", SWOT analysis technique, decision tree technique, "snowball" method, construction of "thought maps")	
18. Group work	✓
19. Tasks on Moodle e-learning platform, https://www.onlinegdb.com , https://codeboard.io , https://repl.it	✓

- Assessment methods to test if learning outcomes have been achieved (please indicate with a tick the appropriate methods for each LO (EU) and/or suggest different methods).

	EU_01	EU_02	EU_03	EU_04	EU_05	EU_06	EU_07	EU_08	EU_09	EU_10	EU_11	EU_12	EU_13	EU_14
Test		✓	✓	✓	✓	✓	✓		✓					
Project	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Report		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Programming task as a homework	✓	✓	✓	✓	✓	✓	✓	✓		✓				
Programming task in a class	✓	✓	✓	✓	✓	✓	✓		✓					



3. Student workload (ECTS credits)

Activity types		Mean number of hours spent on each activity type
Contact hours with the teacher as specified in the study programme		60
Students' self-study*	Preparation for classes	10
	Reading for classes	10
	Preparation of the report	20
	Preparation of the project	20
TOTAL HOURS		120
Total ECTS credits for the course		3

* please indicate the appropriate activity types and/or propose different activities

4. Assessment criteria in accordance with AMU grading scale:

- quiz results (10%),
- class assignments (30%),
- home assignments (30%), as well as
- group project report (15%) and
- group project class presentation (15%)

The grade will be calculated from the mean according to the following scale:

- $\geq 92\%$ very good (bdb; 5,0): excellent knowledge and skills
- $\geq 84\%$ good plus (+db; 4,5): very good knowledge and skills
- $\geq 76\%$ good (db; 4,0): good knowledge and skills
- $\geq 68\%$ sufficient plus (+dst; 3,5): knowledge and skills with minor errors
- $\geq 60\%$ sufficient (dst; 3,0): knowledge and skills with major errors
- $< 60\%$ insufficient (ndst; 2,0): unsatisfactory knowledge and skills