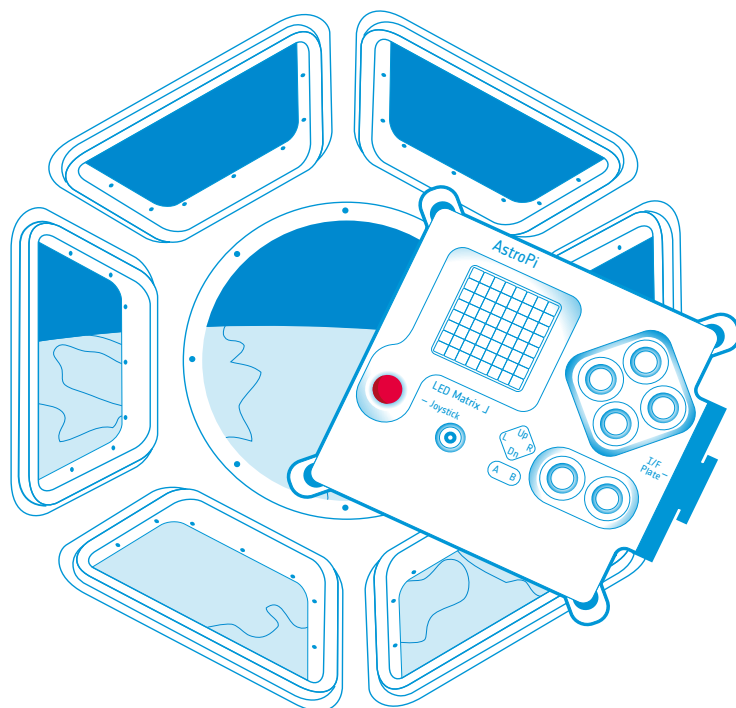


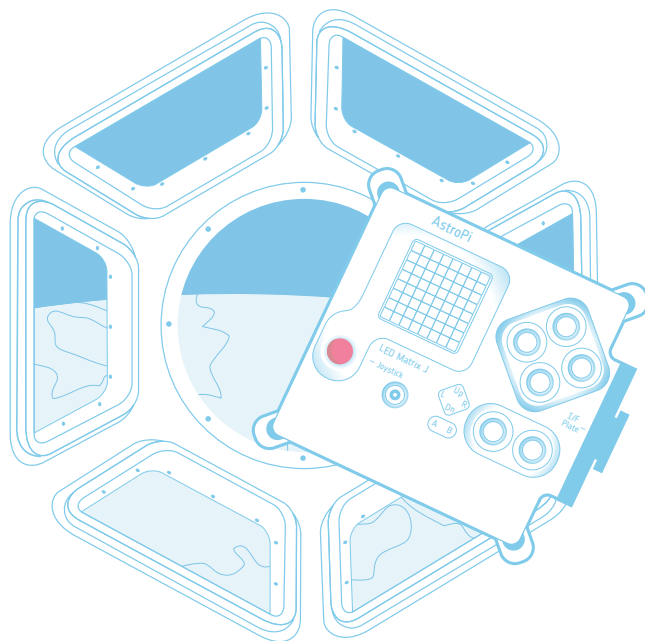


lekcje z kosmosu

→ PIERWSZE KROKI Z MINIKOMPUTEREM ASTRO PI

Raspberry Pi pomaga zrozumieć język programowania





Wstęp	3
Witamy w konkursie European Astro Pi Challenge!	5
Zadanie 1. Poznajemy minikomputer Raspberry Pi	6
Zadanie 2. Poznajemy język Python	8
Zadanie 3. Piszemy quiz naukowy w języku Python	10
Zadanie 4. Rysujemy kształty geometryczne, używając języka Python	12
Przydatne linki	16

→ WSTĘP

Ten przewodnik dla nauczyciela z zadaniami stanowi pierwszą część zestawu materiałów opracowanych przez Biuro Edukacji ESA i jej partnerów w związku z konkursem European Astro Pi Challenge. Ta część zapoznaje ucznia z minikomputerem Raspberry Pi i uczy wykorzystywania jego głównych funkcji. Zadaniom towarzyszy szczegółowe omówienie podstawowych instrukcji kodowania z poglądowymi ilustracjami i zrzutami ekranu.

Materiały i narzędzia

- zestaw Astro Pi
- monitor
- klawiatura USB
- mysz USB

Informacje podstawowe

Wiek: 10–14 lat

Poziom trudności: łatwy

Miejsce: pracownia komputerowa
lub sala lekcyjna

Zakres i treść zajęć

Uczniowie konfiguruje i programują mini-komputer Raspberry Pi. Celem zajęć jest nauka podstawowego języka kodowania.

Uczniowie dowiedzą się

- czym jest minikomputer Raspberry Pi, jakie są jego główne funkcje oraz w jaki sposób go skonfigurować i używać
- jaka jest różnica między sprzętem komputerowym a oprogramowaniem
- czym jest język programowania
- jak programować, wykorzystując język Python
- jak używać komunikatów wejścia i wyjścia, zmiennych, typów danych, instrukcji warunkowych *if* oraz pętli
- jak rysować geometryczne kształty przy użyciu modułu graficznego Turtle z biblioteki Pythona

Zestawienie wynikowe treści

Lp.	Zadanie	Temat	Rezultat	Wymagania
1	Poznajemy minikomputer Raspberry Pi	identyfikacja podzespołów minikomputera Raspberry Pi	zrozumienie, czym jest minikomputer Raspberry Pi; zrozumienie różnicy między sprzętem a oprogramowaniem	brak
2	Poznajemy język Python	napisanie pierwszego programu w języku Python	zapoznanie się z językiem komputera i podstawowymi instrukcjami kodu	brak
3	Piszemy quiz naukowy w języku Python	wprowadzenie podstawowych interakcji z komputerem przy użyciu języka Python	zbieranie danych wejściowych z komputera; poznanie instrukcji warunkowych <i>if</i> i podstawowych instrukcji kodu	brak
4	Rysujemy kształty geometryczne, używając języka Python	wprowadzenie modułu Turtle języka Python	poznanie instrukcji pętli i podstawowych instrukcji kodu	brak

→ WITAMY W KONKURSIE EUROPEAN ASTRO PI CHALLENGE!

Astro Pi to nazwa minikomputera opracowanego przez Raspberry Pi Foundation we współpracy z Brytyjską Agencją Kosmiczną (UK SA) i Europejską Agencją Kosmiczną (ESA). Istnieją dwa bardzo szczególne komputery Astro Pi. Jeden nazywa się Ed, a drugi Izzy. Oba zostały zakwalifikowane do lotu w kosmos i w tej chwili znajdują się na pokładzie Międzynarodowej Stacji Kosmicznej (ISS), pozostając do dyspozycji uczniów takich jak ty.

Podczas tych zajęć zmontujesz swój minikomputer Raspberry Pi i nauczysz się podstaw języka kodowania niezbędnego do programowania Astro Pi. Umożliwi ci to uruchomienie swojego własnego kodu i przeprowadzenie eksperymentu naukowego na stacji ISS.



→ ZADANIE 1. POZNAJEMY MINIKOMPUTER RASPBERRY PI

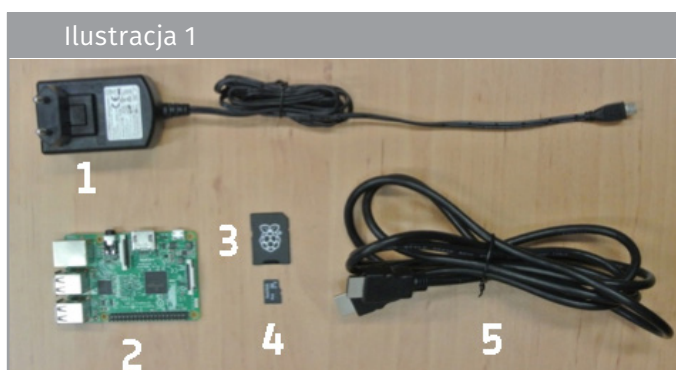
Poza rozmiarem i wyglądem minikomputer Raspberry Pi jest bardzo podobny do znanych nam komputerów. Można go używać do przeglądania Internetu i wysyłania poczty elektronicznej, a także przekształcić w odtwarzacz muzyczny MP3, aby uprzyjemnić czas wolny astronautom i astronautkom na Międzynarodowej Stacji Kosmicznej. Ale można zrobić też dużo więcej! Wykorzystując to narzędzie, można prowadzić eksperymenty naukowe, zbierać i analizować zebrane dane – tak samo jak robią to naukowcy w kosmosie.

W tym zadaniu skonfigurujesz swój minikomputer Raspberry Pi i poznasz jego możliwości.

Ćwiczenie

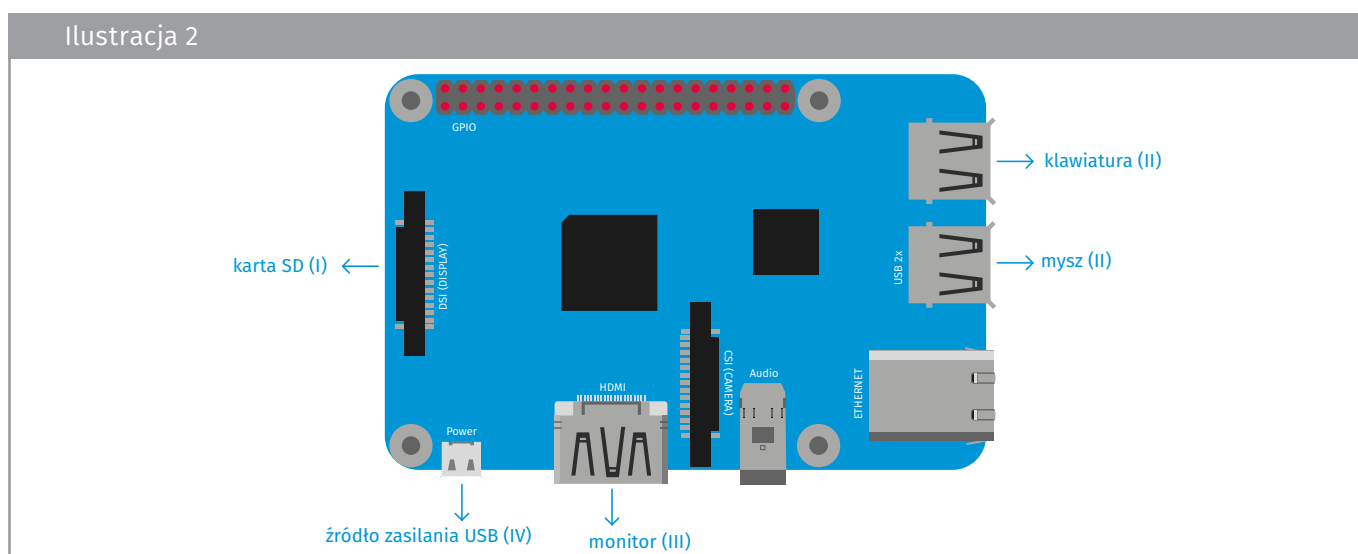
1. Do rozpoczęcia misji Astro Pi będzie ci potrzebny specjalny sprzęt, który znajduje się w zestawie Astro Pi, a także monitor, klawiatura USB i mysz USB. Wszystkie potrzebne materiały i narzędzia przedstawiono na ilustracji 1. Wypisz obok odpowiednich numerów elementy minikomputera Raspberry Pi przedstawione na ilustracji 1: kartę mikro-SD, adapter karty SD, minikomputer Raspberry Pi 3, zasilacz z wtyczką UE, kabel HDMI.

1. _____
2. _____
3. _____
4. _____
5. _____



↑ Elementy minikomputera Raspberry Pi

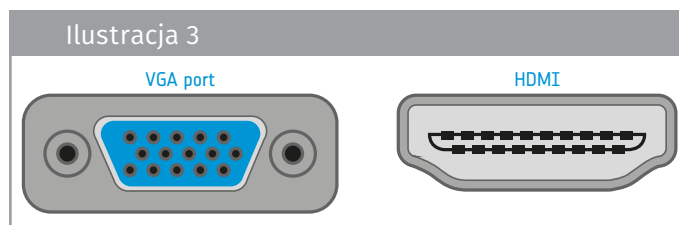
2. Teraz zmontujesz swój minikomputer Raspberry Pi! Postępuj zgodnie z instrukcją opisaną w punktach I–IV, wzorując się na ilustracji 2.



↑ Sposób montażu minikomputera Raspberry Pi

- I. Wyjmij kartę SD z adaptera kart SD. Umieść ją w gnieździe kart SD minikomputera Raspberry Pi. Można to zrobić tylko w jeden sposób.
- II. Podłącz klawiaturę i mysz do odpowiednich portów USB.
- III. Podłącz kabel HDMI od monitora (lub telewizora) do portu HDMI minikomputera Raspberry Pi i włącz monitor.

UWAGA! To może być najtrudniejszy etap montażu minikomputera Raspberry Pi. Upewnij się, że wybrałaś/wybrałeś właściwy port (HDMI, DVI itp.). Jeżeli monitor ma tylko wejście VGA (ilustracja 3), użyj adaptera HDMI-VGA (w zestawie).



↑ Porty VGA i HDMI

- IV. Podłącz zasilacz do portu mikro-USB. Czynność ta spowoduje włączenie i uruchomienie minikomputera Raspberry Pi. Może to zająć ok. 5 minut. Lampka minikomputera powinna migać, a na monitorze powinien pojawić się tekst. Gdy proces uruchamiania minikomputera zostanie zakończony, a tekst na ekranie przestanie się przesuwac, zobaczysz pulpit taki jak ten przedstawiony na ilustracji 4.

3. Zaczynaj korzystać z oprogramowania minikomputera Raspberry Pi. Możesz przykładowo otworzyć przeglądarkę internetową i posurfować po Internecie. Aby to zrobić, kliknij menu główne u góry ekranu (logo Raspberry) i wybierz: *Internet > Chromium Web Browser* (do korzystania z Internetu konieczny jest dostęp do wi-fi). Używając sieci, spróbuj odnaleźć aktualną lokalizację stacji ISS. Zapisz ją.



↑ Pulpit, czyli graficzny interfejs użytkownika

4. Zapoznaj się z oprogramowaniem Raspberry Pi, wybierając różne opcje z menu programowania. Czy teraz potrafisz wyjaśnić różnicę pomiędzy sprzętem i oprogramowaniem? Zanotuj swoje spostrzeżenia.

UWAGA! Jeśli chcesz wyłączyć minikomputer Raspberry Pi, nie wyciągaj wtyczki. Po zakończeniu pracy z minikomputerem wyłącz go, klikając przycisk *Shutdown* w menu głównym.

Ćwiczenie dodatkowe

Nie wszystko jest takie, jakim się wydaje! Zrób listę używanych na co dzień urządzeń cyfrowych, które według ciebie są komputerami. Porównaj swoją listę z listami koleżanek i kolegów. Porozmawiajcie o tym, co decyduje, że urządzenie jest komputerem.

→ ZADANIE 2. POZNAJEMY JĘZYK PYTHON

Aby komunikować się z astronautami i astronautkami na stacji ISS za pomocą minikomputera Raspberry Pi, trzeba mu wydać odpowiednie instrukcje. Tak jak każdy inny komputer zrobi tylko to, do czego zostanie zaprogramowany. Aby zaprogramować komputer, należy się z nim skomunikować, a w tym celu trzeba nauczyć się języka programowania (żeby porozumiewać się ze sobą nawzajem, też musieliśmy nauczyć się języka polskiego).

W tym zadaniu poznasz podstawowe słownictwo języka Python i napiszesz swój pierwszy program.

Czy wiesz, że...

Astronauci i astronautki również muszą się nauczyć różnych języków, zanim będą mogli polecieć w kosmos. Na przykład wszyscy muszą opanować język rosyjski, ponieważ elementy sterujące na statku Sojuz są opisane po rosyjsku. Obecnie Sojuz jest jedynym statkiem kosmicznym, który może transportować ludzi na stację ISS i z powrotem na Ziemię. Słowo „sojuz” w języku rosyjskim oznacza „związek”. Chociaż ten model pojazdu kosmicznego powstał w Związku Radzieckim na początku ery badań kosmosu, jest on używany do dziś – oczywiście po wielu modyfikacjach i unowocześnieniach.

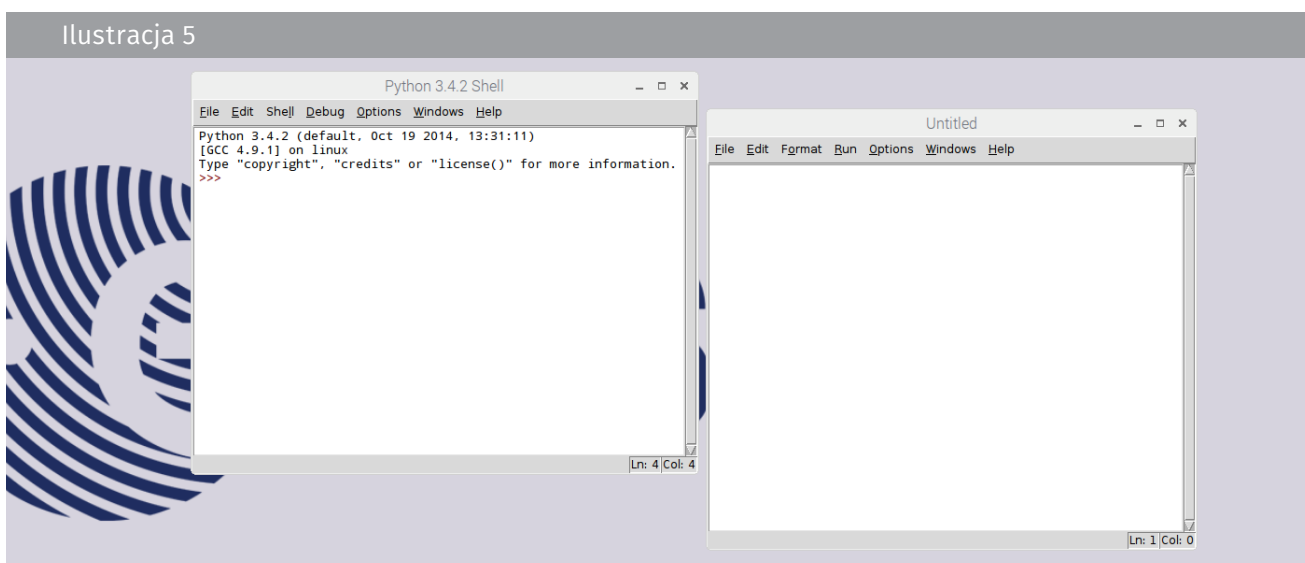


Ćwiczenia

1. Aby rozpocząć używanie języka Python na minikomputerze Raspberry Pi, kliknij menu główne u góry ekranu (logo Raspberry) i wybierz: *Programming > Open Python 3(IDLE)*. Nastąpi otwarcie okna danych wyjściowych powłoki Python Shell (ilustracja 5).

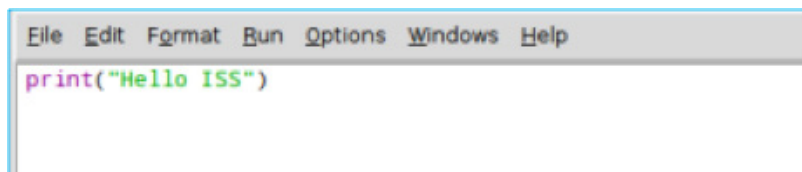
Swoje programy można wpisywać bezpośrednio w oknie powłoki, jednak łatwiej jest pracować w oknie edytora. Aby otworzyć nowe okno, wybierz: *File > New File*. Teraz powinny być otwarte dwa okna – jedno do pisania programu (okno edytora) i jedno do wyświetlania danych wyjściowych (Python Shell), takich jak wyniki czy komunikaty o błędach. Upewnij się, że widzisz na ekranie obydwa okna.

Ilustracja 5



↑ Okno powłoki Python Shell (po lewej) i edytora (po prawej)

2. Teraz możesz rozpocząć kodowanie w języku Python. Wpisz w oknie edytora poniższy kod.

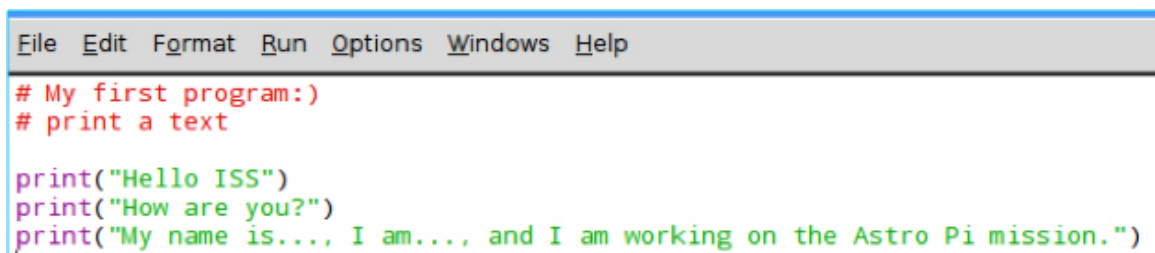


```
File Edit Format Run Options Windows Help
print("Hello ISS")
```

3. Aby zobaczyć efekt działania swojego kodu, wybierz: *Run > Run Module*. Program poprosi o zapisanie kodu. Zatytułuj swój plik, zapisz jego nazwę i uruchom moduł. Co się stało?

UWAGA! Pamiętaj o regularnym zapisywaniu swojej pracy! Aby zapisać plik, nad którym pracujesz, wybierz: *File > Save*.

4. Spróbuj przedstawić się astronautom i astronautkom mieszkającym na stacji ISS. Skopiuj poniższy kod do swojego pliku i wpisz treść wiadomości, którą chcesz im przekazać. Wiadomość powinna być w języku angielskim, żeby astronauta i astronautki mogli ją zrozumieć.



```
File Edit Format Run Options Windows Help
# My first program:)
# print a text

print("Hello ISS")
print("How are you?")
print("My name is..., I am..., and I am working on the Astro Pi mission.")
```

Gratulacje! Udało ci się właśnie stworzyć swój pierwszy program w języku Python.

UWAGA! Komputer będzie próbował zrozumieć wszystko, co napiszesz w oknie. Jeżeli chcesz napisać krótki komentarz tylko dla siebie, wstaw przed notatką znak *#*. Oznacza on tryb komentowania. Tak wprowadzony komentarz nie zostanie przetłumaczony na język maszynowy. Komputer po prostu ignoruje tekst wpisany po tym znaku.

→ ZADANIE 3. PISZEMY QUIZ NAUKOWY W JĘZYKU PYTHON

Do tej pory dawałaś/dawałeś komputerowi proste instrukcje. Teraz sprawisz, że komputer będzie dawał pytania i reagował na twoje odpowiedzi.

Ćwiczenia

1. Otwórz nowe okno Python 3 znanim ci już poleceniem: *File > New File* i wpisz poniższy kod.

```
File Edit Format Run Options Windows Help
# whenever a word or number follows the '=' symbol, it is called a variable.
# variables are used to store information that the computer must remember.

orbits = input (" How long does it take the ISS to orbith the Earth?")
```

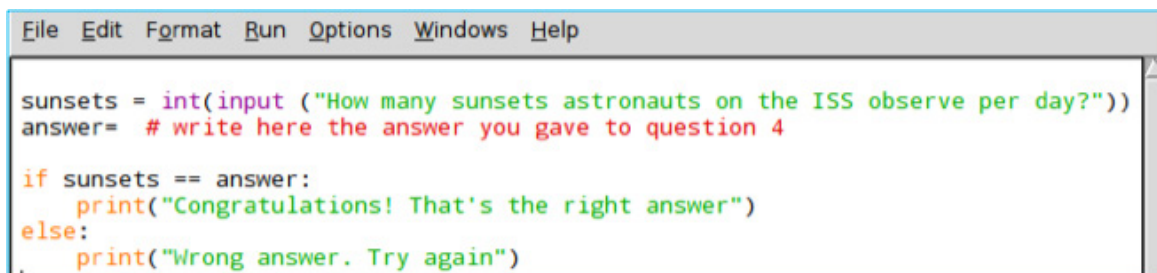
2. Wybierz: *Run > Run Module*, aby wykonać wprowadzony kod. Co się stało?

3. Funkcja *input ()* w języku Python służy do pobierania danych od użytkownika. Czy wiesz, ile czasu potrzebuje stacja ISS, aby okrążyć Ziemię? Wpisz to pytanie w powłoce Python Shell.
4. Powłokę Python Shell można też wykorzystać jako kalkulator. Zapoznaj się z poniższymi przykładami.

```
File Edit Shell Debug Options Windows Help
Python 3.4.2 (default, Oct 19 2014, 13:31:11)
[GCC 4.9.1] on linux
Type "copyright", "credits" or "license()" for more information.
>>> 12+12
24
>>> 12*2
24
>>> 12/2
6.0
>>> 12*2-12
12
```

Ile zachodów słońca widzą astronauta na stacji ISS w ciągu jednego dnia?
Spróbuj odpowiedzieć na to pytanie, używając powłoki Python Shell do wykonania potrzebnych działań matematycznych. Objaśnij przeprowadzone obliczenia.

5. Napiszesz teraz quiz naukowy, używając minikomputera Raspberry Pi. W quizie trzeba będzie wybrać prawidłową lub nieprawidłową odpowiedź. W języku Python możemy to zrealizować przy użyciu instrukcji wyboru, nazywanych czasami instrukcjami *if*. Otwórz nowy plik i wpisz poniższy kod.



```
File Edit Format Run Options Windows Help
sunsets = int(input ("How many sunsets astronauts on the ISS observe per day?"))
answer= # write here the answer you gave to question 4

if sunsets == answer:
    print("Congratulations! That's the right answer")
else:
    print("Wrong answer. Try again")
```

Musisz dokończyć kod, wpisując odpowiedź. Jak myślisz, co zrobi program? Uruchom moduł, aby się przekonać.

UWAGA! Zapis *int* przed instrukcją *input* oznacza w języku Python, że odpowiedź użytkownika będzie liczbą całkowitą. Jeżeli jakaś informacja ma być traktowana w języku Python jako liczba, należy poinformować o tym komputer odpowiednią instrukcją. Oprócz liczb całkowitych można wykorzystywać także liczby zmiennoprzecinkowe. Liczby zmiennoprzecinkowe to liczby zapisane w postaci ułamka dziesiętnego – z miejscami dziesiętnymi, np. 5,1 lub 10,99.

6. Quiz można nieco ułatwić. Tuż przed wierszem *else* dodaj poniższe polecenie i ponownie uruchom moduł. Dopisz komentarz wyjaśniający, jakie zadanie spełnia słowo kluczowe *elif*.

```
# _____
elif sunsets >13 and sunsets <16:
    print("Almost there! Try again")
```

Ćwiczenie dodatkowe

Instrukcje warunkowe *if* mogą sprawdzać prawdziwość wielu warunków. W tym celu wystarczy użyć wymaganej liczby instrukcji *elif*. Napisz program, który poprosi użytkownika o podanie liczby z przedziału od 45 do 60. W reakcji na odpowiedź niech informuje, czy podana przez użytkownika liczba mieści się w tym przedziale, czy jest zbyt duża, czy też zbyt mała.

→ ZADANIE 4. RYSUJEMY KSZTAŁTY GEOMETRYCZNE, UŻYWAJĄC JĘZYKA PYTHON

Język Python zawiera wiele modułów, które możesz wykorzystać w swoich programach. Moduł można porównać do książki z instrukcjami, która objaśnia, w jaki sposób coś zrobić. Takie moduły są przechowywane przez język Python w bibliotece.

W tym zadaniu nauczysz się używać modułu Turtle (żółw) języka Python do rysowania kształtów geometrycznych, a jednocześnie poznasz kolejne elementy języka Python, które będą przydatne w twojej misji.

Ćwiczenia

1. Otwórz nowe okno Python 3 i wpisz poniższy kod. Zapisz go i uruchom moduł.

```
File Edit Format Run Options Windows Help
# to begin using Turtle you need to import the Turtle from the Python library
from turtle import Turtle, Screen

screen = Screen() # creates the window wich will display the turtle drawings
turtle = Turtle() # creates a turtle. The variable (ou noun) turtle is made to refer to it.

turtle.shape("turtle") # gives the turtle shape
```

2. Właśnie utworzyłeś/utworzyłaś żółwia, który pozwoli ci tworzyć rysunki i kształty! Wydajmy więc naszemu żółwiowi kilka instrukcji.

- 2.1. Dodaj do swojego kodu następujący wiersz:

```
turtle.forward(100)
```

Co się stało z żółwiem?

- 2.2. Dodaj kolejną instrukcję:

```
turtle.right(90)
```

Zmień liczby w nawiasach. Spróbuj zastąpić słowo *right* (prawo) słowem *left* (lewo). Jak na te instrukcje reaguje żółw?

UWAGA! Wartość 90 zapisana w nawiasie w instrukcji *turtle.right(90)* jest podana w stopniach.

- 2.3. Dodaj kolejną instrukcję, aby dalej poruszać żółwiem:

```
turtle.forward(100)
```

Co się stało z żółwiem?

2.4. Co trzeba dodać do kodu, aby dokończyć kwadrat? Wpisz w ramce końcowy fragment kodu potrzebny do tego, aby żółw narysował kwadrat.

3. Żółwiowi można dawać wiele różnych instrukcji. Przeanalizuj poniższy kod.

```
File Edit Format Run Options Windows Help
from turtle import Turtle, Screen

screen = Screen()
turtle = Turtle()

turtle.shape("turtle")

turtle.forward(100)
turtle.right(120)
turtle.forward(100)
turtle.right(120)
turtle.forward(100)
```

3.1. Jaki kształt geometryczny twoim zdaniem narysuje żółw? Zapisz swoją odpowiedź.

3.2. Zapisz i uruchom kod. Czy miataś/miałaś rację?

3.3. W zamieszczonym powyżej kodzie powtórzono kilka wierszy. Takie powtarzanie w informatyce jest czasochłonne i niepotrzebne. Zamiast wpisywać te same wiersze kodu wiele razy, można napisać je raz i dodać instrukcję ich powtarzania. W języku Python istnieją dwa sposoby powtarzania instrukcji. Są to pętla *while* oraz pętla *for*.

- 3.3.1. Otwórz nowe okno Python 3 i wpisz poniższy kod. Kod wewnątrz pętli (tj. kod, który jest zapisany z wcięciem, czyli przesunięty w prawo) to zestaw instrukcji, które program będzie powtarzał. Zapisz i uruchom program.

```
File Edit Format Run Options Windows Help
from turtle import Turtle, Screen

screen = Screen()
turtle = Turtle()

turtle.shape("turtle")

turtle.forward(100)
turtle.right(120)
turtle.forward(100)
turtle.right(120)
turtle.forward(100)
```

Co nastąpiło?

- 3.3.2. W tym przykładzie użyto pętli *for*. Wpisz i uruchom poniższy kod, aby zobaczyć, co się stanie.

```
File Edit Format Run Options Windows Help
from turtle import Turtle, Screen

screen = Screen()
turtle = Turtle()

turtle.shape("turtle")

for i in range(3):
    turtle.forward(100)
    turtle.right(120)
```

UWAGA! Przypatrz się zagnieżdżeniu wciętych instrukcji w pętlach *while/for*. Język Python wykorzystuje wcięcia do zaznaczania bloków kodu w pętlach lub instrukcjach warunkowych *if*. Każdy poziom wcięcia tworzą cztery spacje, lecz można je wstawić jednym naciśnięciem klawisza Tab. Podczas kopiowania i wklejania kodu nie można zapominać o wcięciach.

- 3.3.3. Którą pętlę należy wybrać, aby stworzyć rysunek? Uzasadnij swój wybór i wyjaśnij, na czym polega różnica pomiędzy obydwooma typami pętli.
-
-

Ćwiczenie dodatkowe

Wiesz już, jak się poruszać do przodu, w prawo i w lewo, a także jak wykonywać powtórzenia przy użyciu pętli. Czas na kreatywność. Czy potrafisz stworzyć rysunek Paxiego – maskotki ESA? Jakie kształty geometryczne rozpoznajesz w postaci Paxiego?



Kolejne kroki

Zmontowałeś/zmontowałaś już minikomputer Raspberry Pi i poznałeś/poznałaś podstawy programowania w języku Python. W kolejnym kroku nauczysz się używać matrycy LED nakładki Sense HAT do komunikowania się z astronautami i astronautkami na Międzynarodowej Stacji Kosmicznej.

→ PRZYDATNE LINKI

- seria trzech lekcji przeznaczonych dla uczniów szkół ponadpodstawowych wprowadzających programowanie w języku Python (autor: Nichola Wilkin; UK National STEM Learning Centre):
<https://www.stem.org.uk/elibrary/resource/35836>
- interaktywne środowisko języka Python:
<https://trinket.io/python>
- materiały edukacyjne Raspberry Pi Foundation, zawierające więcej informacji o module Turtle i programowaniu:
<https://www.raspberrypi.org/learning/turtley-amazing/worksheet/>
<https://www.raspberrypi.org/learning/turtle-snowflakes/>

Tytuł oryginału: *Teach with space: Getting started with Astro Pi | T05.1*

www.esa.int/education

Opracowane przez Biuro Edukacji ESA we współpracy z Raspberry Pi Foundation
i oddziałami Esero w Polsce i Wielkiej Brytanii

Copyright 2017 © European Space Agency