

Automatyzacja nudnych zadań w Pythonie

Rozdział 3

1. Dlaczego istnienie funkcji jest korzystne dla programu?

Odp.: Ponieważ zapobiega duplikacji kodu w programie co przekłada się na mniejszą ilość błędów.

2. Kiedy nastąpi wykonanie kodu funkcji: w chwili jej zdefiniowania, czy w momencie wywołania?

Odp.: W momencie wywołania

3. Jakie polecenie służy do utworzenia funkcji?

Odp.: `def nazwa_funkcji():`

4. Jaka jest różnica między funkcją i wywołaniem funkcji?

Odp.: Funkcja składa się z polecenia `def` oraz kodu w jego klauzurze. Wywołanie funkcji to operacja przenosząca wykonywanie programu do tej funkcji oraz powodująca obliczenie wartości zwrotnej funkcji.

5. Ile zasięgów globalnych i lokalnych istnieje w programie Pythona?

Odp.: Istnieje jeden zasięg globalny i zasięg lokalny który jest tworzony w momencie wywołania funkcji.

6. Co się stanie ze zmiennymi w zasięgu lokalnym, gdy zakończy się działanie funkcji?

Odp.: Zostają usuwane.

7. Co to jest wartość zwrotna? Czy wartość zwrotna może być częścią wyrażenia?

Odp.: Wartość zwrotna to wartość zwracana przez funkcję i może być częścią wyrażenia.

8. Jeżeli funkcja nie ma polecenia `return`, jaka będzie wartość zwrotna

wywołania tej funkcji?

Odp.: None

9. Jak można zmusić zmienną w funkcji do odwołania się do zmiennej globalnej?

Odp.: Poleceniem global

10. Jaki jest typ danych dla wartości None?

Odp.: NoneType

11. Na czym polega działanie polecenia import areallyourpetsnamederic?

Odp.: Na dołączeniu modułu areallyourpetsnamederic do programu.

12. Jeżeli masz funkcję o nazwie bacon() w module spam, w jaki sposób będziesz wywoływał tę funkcję po zaimportowaniu wymienionego modułu?

Odp.: spam.bacon()

13. W jaki sposób można nie dopuścić do awarii programu po wystąpieniu w nim błędu?

Odp.: Za pomocą try i except

14. Co należy umieścić w klauzulach try i except?

Odp.: W klauzuli try należy umieścić kod które spowoduje błąd, a w klauzuli except umieścić kod który ma się wykonać w przypadku błędu.