Theory: Variables

7 minutes 9 / 10 problems solved

Start practicing

You can use a programming language like Python to do calculations or to work with constant values like strings. Is it enough for you, though? When writing real programs, you usually need to **store values** or evaluation results in computer memory.

§1. What is a variable?

Variable is a named place where you can store some value and access the value later. Imagine a *box* where you keep something. That's a variable.

For example, you calculate something and would like to reuse the formula for some other numbers. In this case, you operate only these "boxes".

In general, it's a good practice to give a variable a name that describes its content.

§2. Defining a variable and assigning values

You can keep almost anything in variables just assigning the new value for a named variable with an equal sign. Also, following PEP 8, one space before and after the assignment sign is considered good practice.

```
day_of_week = "Monday"
```

Now you have a string typed value `"Monday"` stored in computer memory. You can retrieve the value by calling the variable name.

```
print(day_of_week)  # Monday
```

Now, `day_of_week` stores a value of the `str` type.

```
print(type(day_of_week))  # <class 'str'>
```

You can always assign a new value to a defined variable:

```
day_of_week = "Tuesday"
```

Now, you will retrieve another value:

```
print(day_of_week)  # Tuesday
```

It is possible to assign the value of one variable to another variable:

```
a = 10
b = a  # b is 10
```

If you haven't defined a variable within the scope of your code, you'll see an error:

```
print(month_name)  # NameError: name 'month_name' is not defined
```

Python allows you to assign values of different types to the same variable. Let's assign the string name of a month to a variable and print its type.

```
month = "December"
print(type(month))  # <class 'str'>
```

Now, let's assign the number of this month to the variable and print its type again.

Current topic:

- ✓ Variables 17★ (Stage 1)

Topic depends on:

- ✓ Basic data types 17★ (Stage 1)

Topic is required for:

- ✓ Type casting 12★
- ✓ Naming variables 17★ (Stage 1)
- ✓ Taking input 17★ (Stage 1)
- ✓ Boolean logic 16★ (Stage 1)
- ✓ List 13★ (Stage 1)
- ✓ Dictionary 6★ (Stage 1)
- ✓ Invoking a function 16★ (Stage 1)
- ✓ Exceptions 3★ (Stage 1)

Table of contents:

↑ Variables

§1. What is a variable?

§2. Defining a variable and assigning values

§3. Naming rules

Feedback & Comments

```
month = 12
print(type(month))  # <class 'int'>
```

This works because Python is a language with *dynamic typing*.

> Please, do not overuse it! If your code is quite long, you might forget that you have changed the type. And this is the breeding ground for errors!

§3. Naming rules

As we mentioned above, each variable has a specific name that distinguishes it from other variables. There are some rules for naming variables that you should follow:

- names are case-sensitive (`month` is not the same as `Month`);
- a name begins with a letter or an underscore, followed by letters, digits, and underscores (e.g. `user_name`, `score1`, `_count`);
- name cannot start with a digit (e.g. `2q` is not a valid name);
- a name must not be a keyword.

> **Do not break these rules**; otherwise, your program will not work.

Report a typo

2297 users liked this theory. **14** didn't like it. **What about you?**

Start practicing

Comments (16) Hints (0) Useful links (2) Show discussion