体験入学
Pythonプログラム

2020年8月22日

コンピュータ教育学院

1. Pythonを学ぶ前に知っておきたいこと

学習を始める前に、Pythonというプログラミング言語が得意とする分野、またPythonのバージョンなど基礎知識について解説します。

1.1. Pythonの強みが活かされている分野と実績

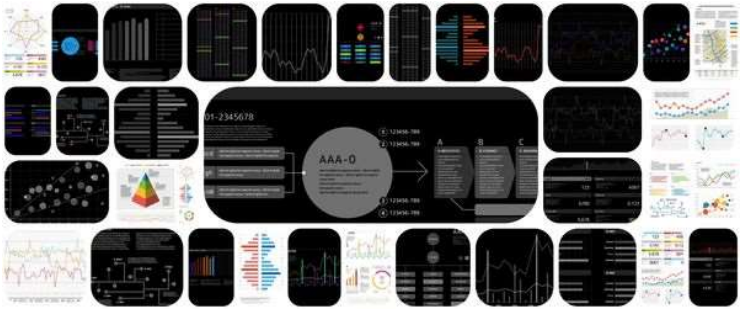
1.1.1 人工知能(AI)

学習・推論・判断などの人間の知能のはたらきをコンピューターが行い、また人口知能自らが考える力を備えています。
Python実績例：ソフトバンク社の開発したPepper(感情エンジン搭載ロボット)

1.1.2 IoT(アイオーティー)

身の回りのあらゆるものにコンピューターが仕組まれ、インターネットとつながります。一例として、車の自動運転や、家屋内のエネルギー消費の最適化などがあります。
Python実績例：Nest Thermostat(屋内自動温度管理)、PlushCare(遠隔医療サービス)

1.1.3 データ解析や分析ツール



ビッグデータ(巨大なデータ群)の中から価値のある情報を見つけ出します。例えば、ユーザーの性別・職業・年齢と購入商品を分析して新製品を開発するなど、個人に特化したサービスの提供や、企業の業務効率化に役立ちます。
Python実績例：Tableau(データのビジュアル分析ツール)

1.1.4 Webアプリケーションの開発

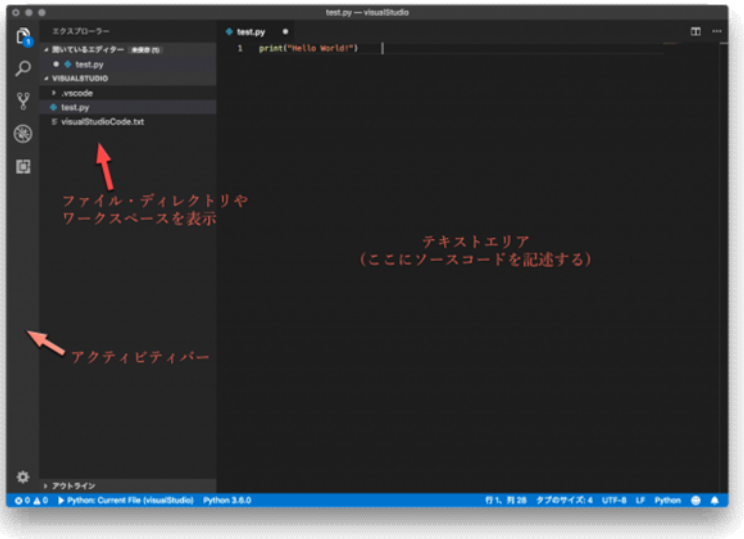
有名なアプリケーションもPythonで作られています。Pythonは各種モジュールが充実していることもあり、容易にメンテナンスを行える機能を迅速に作れるため、大規模なWebアプリケーションへの対応が可能です。
Python実績例：Youtube, Instagram, Dropbox

そして近年、上記に挙げたような分野において、Pythonのスキルが活かせる仕事や求人が急増しています。

2. Pythonを動かす環境

2.1　VisualStudioCodeとは

Visual Studio Codeとは、Microsoftが提供する高機能なエディタです。略称はVSCodeと言います。こちらはIDE(統合開発環境)のVisual Studioと異なりエディタではありますが、コーディングはもちろんのことデバッグ、Git連携、機能の追加までできてしまう優れものです以下の画像のようなUIでVSCodeは使うことができます。



2.2　他のエディタとの違い

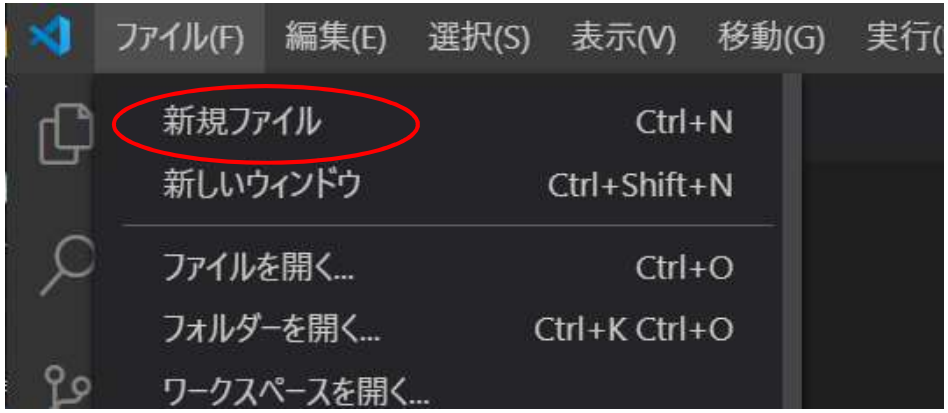
VSCodeの他エディタとの違いは、多様な機能にあります。まずエディタとしての機能としては検索、置換、Grep等の検索関連の機能をはじめ、基本的な機能は一通り備わっているかと思います。
それらの機能に加え、先述の通りデバッグやGit連携ができたり、拡張機能で新しい機能をインストールしてくることで追加することもでき、自分の使い方にあったカスタマイズが可能です。
また、多くのプログラミング言語、マークアップ言語に対応しています。関数等に色付けしてくれたり、コーディングするにあたって補助となる機能もバッチリです。
さらにUMLやマークダウンのプレビュー機能もついていて、資料を作成する際にも活用できます。つまりエディタの機能に加え、開発に必要な機能が上乗せされたものがVSCodeです。
機能面ではAtomにも似ていますが、VSCodeは動きが早い点が魅力です。

3. とりあえずPythonを動かしてみよう

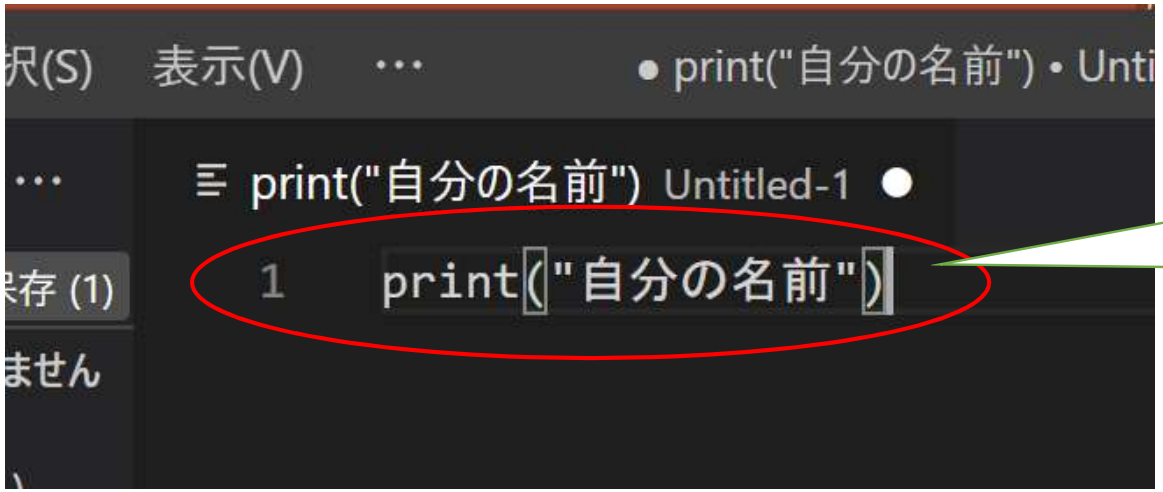
3.1 VisualStudioCodeを起動

デスクトップのアイコンをダブルクリック

3.2 新規ファイル作成

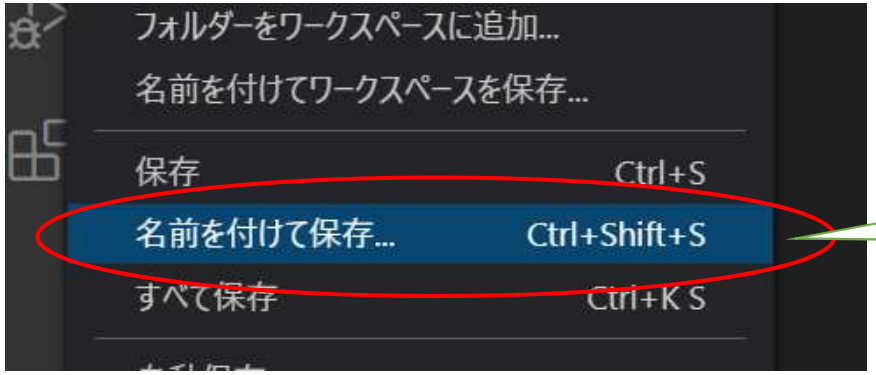


3.3　プログラム入力（キーボード入力）



実際に自分の名前をいれる

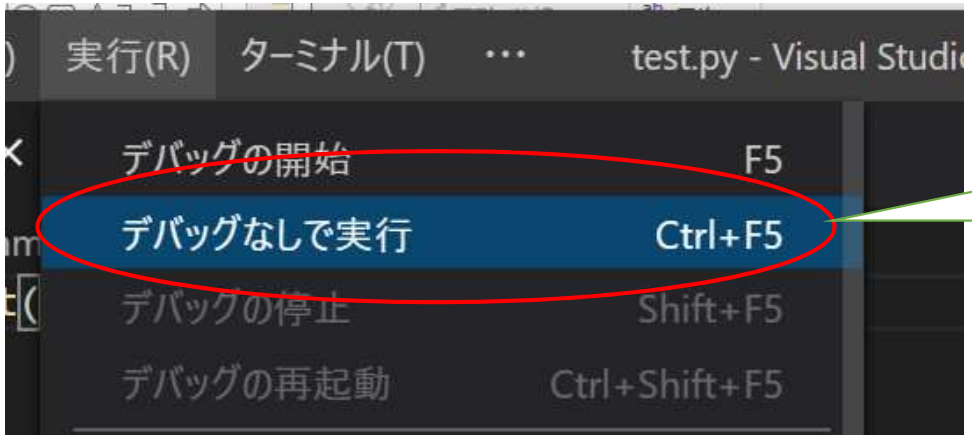
3.4　ファイル保存（デスクトップにファイル保存）



ファイル
→名前を付けて保存

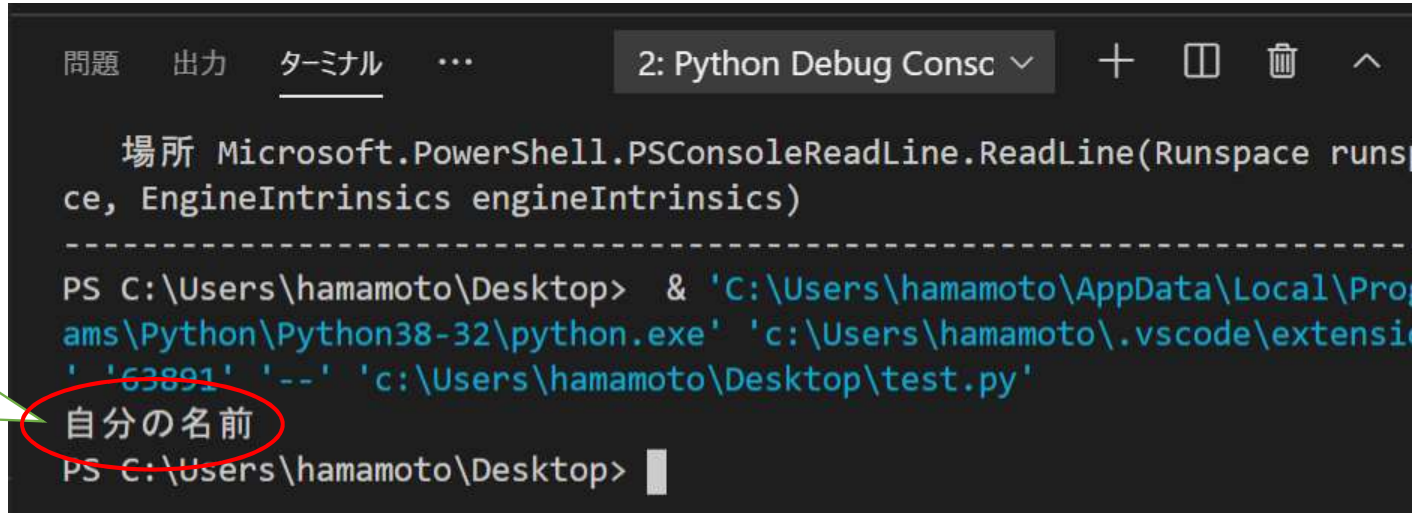
ファイル名：test.py

3.5　実行する



実行
→デバッグなしで実行

3.6　実行結果確認



ターミナルに表示されること

4. Pythonプログラムの基本①

4.1 文字列を出力してみよう

```
print('Hello Python')
```

4.2 数値を出力してみよう

```
print(3)
print(3 + 7)
print(7 - 3)
```

4.3 文字列と数値の違い

```
print(3 + 5)
print('3 + 5')
```

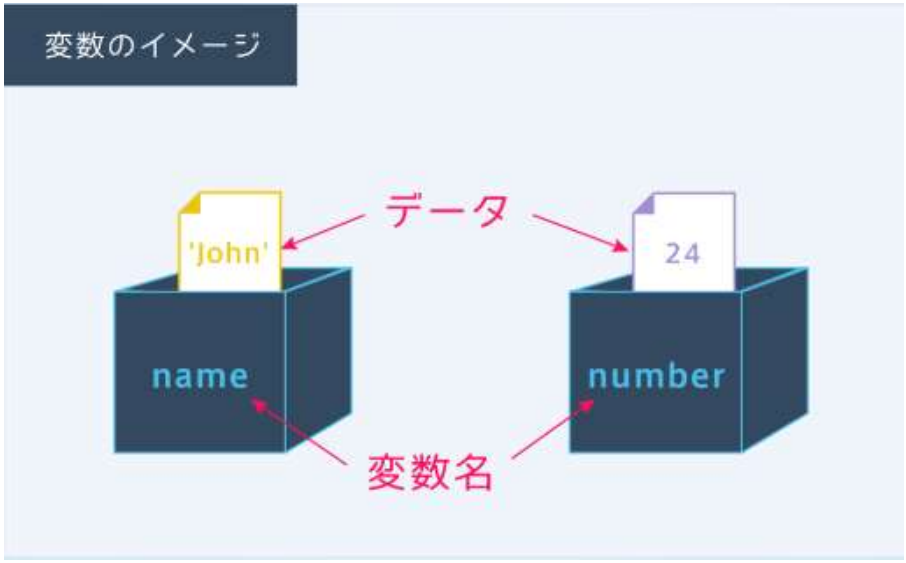
4.4 計算してみよう

```
print(3)
print(3 + 7)
print(7 - 3)
```

4.5 その他の計算

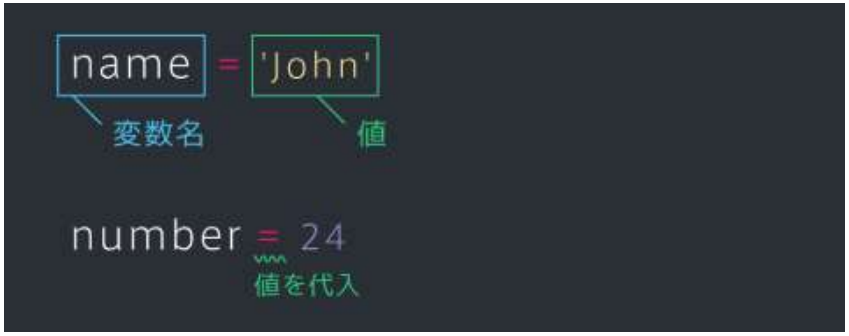
```
print(3 * 7)
print(3 / 2)
print(7 % 3)
```

4.6 変数とは？



変数とは、データ（値）を入れておく箱のようなものです。
この箱（変数）に名前（変数名）をつけることで、その名前を用いて変数に値を入れることや、変数から値を取り出すことができます。

4.7 変数の定義



4.8 変数の値を取り出してみよう

```
name = 'John'

print(name)
        変数 name の値

print('name')
        「name」という文字列
```

4.9変数を使う意義①

```
apple_count = 3
変数名が何を表しているか分かりやすい

apple_price = 100

total_price = apple_count * apple_price

print(total_price)  # 結果：300
```

変数を使うメリットの1つは、データに名前をつけることで、扱っているデータの中身が何を表しているのかが明確になることです。

4.10 変数を使う意義②

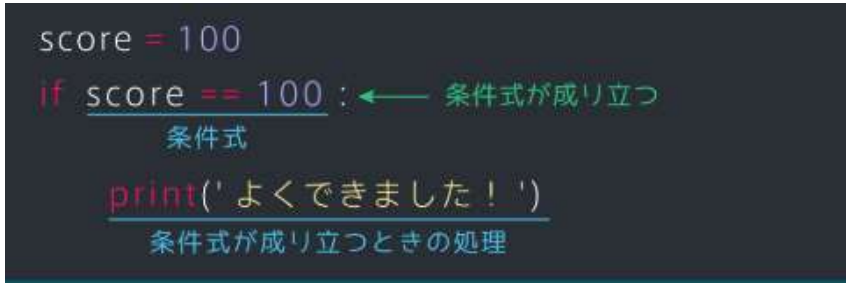
変数を使うことには、他にも以下のようなメリットがあります。
・同じデータを繰り返し利用することができる
・変数の値に変更が必要になった場合、変更する箇所が1箇所で済む

4.11 変数の値を更新する①

4.12 変数の値を更新する②

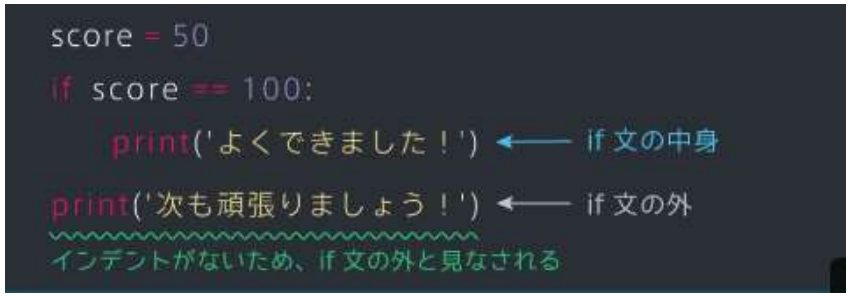
5. Pythonプログラムの基本②

5.1 条件分岐(if)



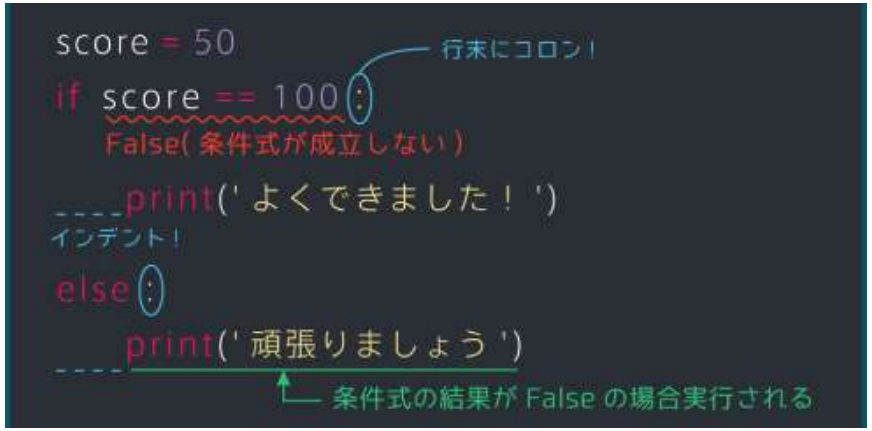
if文を用いると「もし○○ならばXXを行う」という条件分岐が可能になります。
if文は図のように書きます。ifの後に条件式を指定し、その条件が成り立つときに実行する処理を次の行に書きます。

5.2 インデント



if文の条件式が成立した時の処理を書くときには、インデント（字下げ）をします。
図のように、処理がif文の中にあるかどうかはインデントによって判別されます。条件が成立したときにif文の中の処理が実行されます。
Pythonではコードの見た目（インデント）がそのままプログラムの動作に影響するので、インデントに気をつけましょう。

5.3条件に合致しない時の処理(else)



6. Pythonプログラム演習

6.1 簡単な数当てゲームを作成する

```
import random

a = random.randint(0, 9)
print(a)

b = int(input("数を入力してください>"))
if a == b:
    print("あたり")
else:
    print("はずれ")
```

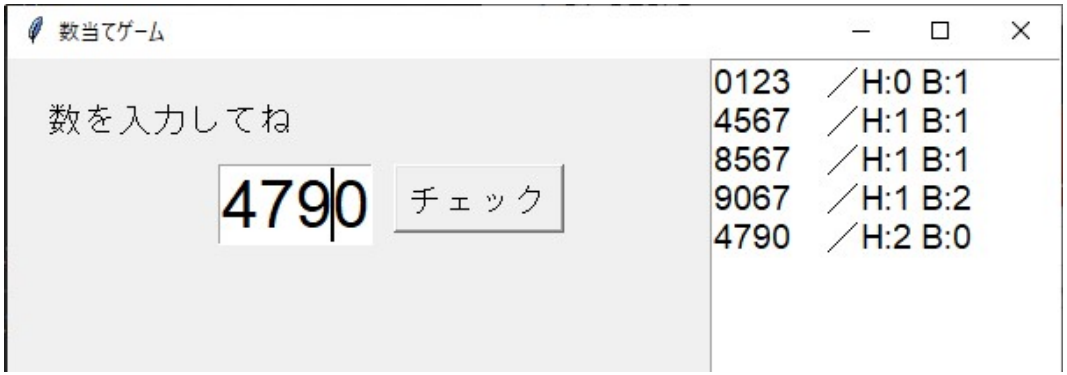
ランダムな数を作る

人間が入力する

あたり、はずれを判断する

6.2 数当てゲームの完成形を実行してみる

ゲーム名：ヒットアンドブロー
実行ファイル：hit-and-blow.py



H:ヒットの数
B:ブローの数
※ブローとは「位置は正しくないがその数字が含まれる」