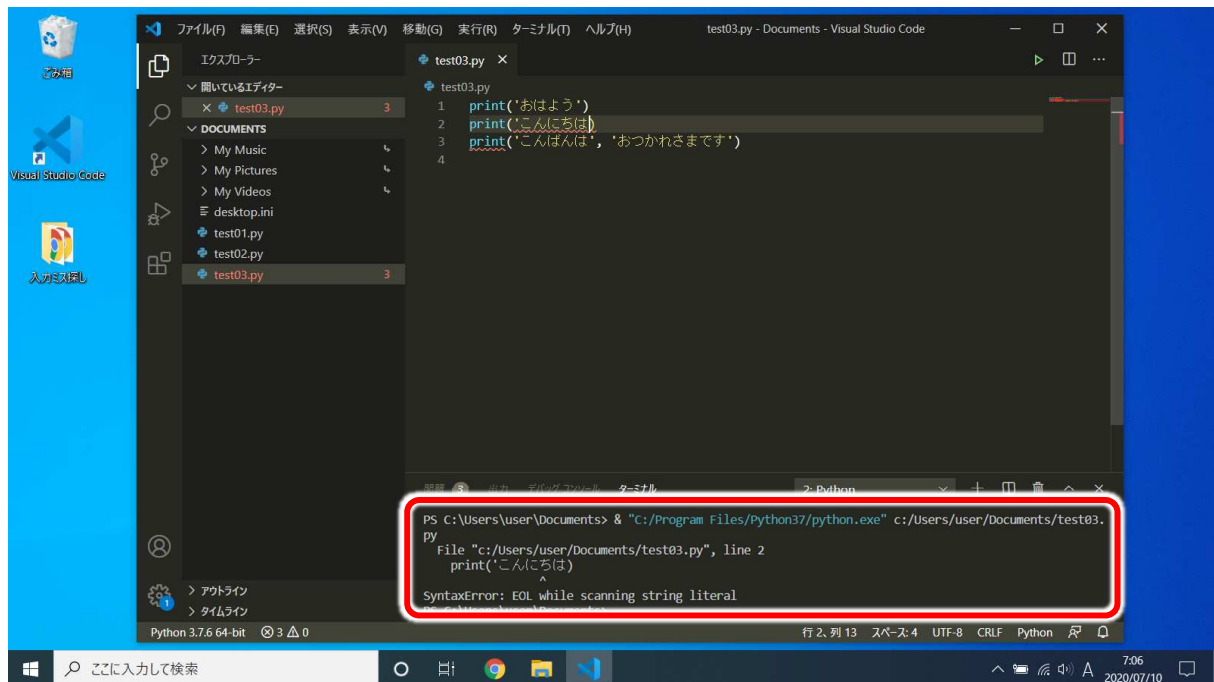


## エラーと対処方法

主なエラーと対処方法をまとめました。

### 1 主なエラーと対処方法

エラーは、次のように英語のメッセージで示されます。



#### (エラー例)

```
PS C:\Users\user\Documents> $ "C:/Program Files/Python37/python.exe"
c:/Users/user/Documents/test03.py
```

```
File "c:/Users/user/Documents/test03.py", line 2
```

```
print('こんにちは')
```

```
^
```

```
SyntaxError: EOL while scanning string literal
```

違反が起きている行番号

エラーが検出された最初の位置を示す矢印

エラー・メッセージには、違反の起きている行番号やエラーが検出された最初の位置を示す小さな「矢印」があります。これは、矢印の場所か、その周辺にプログラムの書き方のミスがあることを意味します。

エラーは、「**構文エラー**」と「**例外**」の2種類に分かれます。

## 1-1 構文エラー

構文エラーとは、プログラムの記述が Python のルールにあっていないことによるエラーです。

### SyntaxError

余計なカッコ()やクォーテーション"" ''、コロン:がある、あるいは足りないことなどによる構文エラー

( Syntax とは「構文、文法」を意味する英語 )

(対処方法)

矢印周辺にあるルールにあっていない命令文を見つけ、修正します。

- 余計なカッコやクォーテーション、あるいは囲み忘れ、はありませんか？
- if 文や for 文の最後にコロンを付け忘れ、余計なコロンはありませんか？
- 日本語入力で全角のカッコやクォーテーション、コロンなど記号を記述していませんか？ (エラー・メッセージに invalid character in identifier(「識別子が無効な文字」を意味する英語)が表示されていませんか？)

(エラー例)

```
File "c:/Users/user/Documents/test03.py", line 2
    print('こんにちは')
          ^
```

SyntaxError: EOL while scanning string literal

### IndentationError

字下げ(インデント)が正しくないことによる構文エラー

( Indentation とは「( 行の ) 字下げ」を意味する英語 )

(対処方法)

矢印周辺にある字下げ位置のミスを見つけ、修正します。

- if 文や for 文の次行の字下げ忘れはありませんか？
- 字下げの必要のない文の先頭に不要なスペースはありませんか？

(エラー例)

```
File "c:/Users/user/Documents/test15.py", line 3
    print("アイスクリームを用意しましょう")
    ^
```

IndentationError: expected an indented block

## 1-2 例外

例外とは、プログラムに記述した文や式が構文的に正しくても、実行しようとしたときにエラーが発生するものです。常に致命的というわけではありません。

NameError
そのような名前で定義されている変数や関数が見つからなかった場合のエラー。 (Name とは「名前」を意味する英語)
(対処方法) 違反が起きている行あるいは周辺から、変数名や関数名のスペル・ミスを見つけ、修正します。
<ul style="list-style-type: none"><li>● 変数、関数のスペルにミスはありませんか？</li><li>● プログラムで使っている変数名が途中で違うスペルになっていませんか？</li></ul>
(エラー例) File "c: /Users/user/Documents/test01.py", line 1, in <module> prnit('Hello world') NameError: name 'prnit' is not defined

ValueError
型は合っているが値が適切でない場合のエラー。 (Value とは「値」を意味する英語)
(対処方法) 違反が起きている行から、値が適切でない箇所を見つけ、修正します。 例えば float() は文字列(str)を浮動小数点(float)に変換するが、元の文字列が変換可能な値でないとエラーになる。
<ul style="list-style-type: none"><li>● 関数に渡す値は適切ですか？ (たとえば、小数型へ変換する float() 関数に数値にならない文字列を渡す float('abc') など)</li></ul>
(エラー例) File "c: /Users/user/Documents/test15.py", line 1, in <module> temp = int(input("気温を入力してください")) ValueError: invalid literal for int() with base 10: 'abc'

## TypeError

不適切な型に対して演算や組み込み関数による処理が行われた場合のエラー。

(Type とは「型」を意味する英語)

### (対処方法)

違反が起きている行から、不適切な型を使った演算や処理をみつけ、修正します。

- 演算する変数や値の型が異なっていませんか？  
(たとえば、文字列型の値 'abc' と整数型の値 10 の演算など)
- 関数に渡す変数や値の型は適切ですか？  
(たとえば、小数型へ変換する float() 関数に配列を渡す float([1, 2, 3]) など)

### (エラー例)

```
File "c: /Users/user/Documents/test25.py", line 3, in <module>
    print(bento_price + student_number)
TypeError: can only concatenate str (not "int") to str
```

## IndexError

配列(リストなど)に格納された値を[インデックス]で取得するときに、範囲外(配列の要素数ではない)のインデックスを指定した場合のエラー

(Index とは「索引」を意味する英語)

### (対処方法)

違反が起きている行から、インデックスのミスを見つめ、修正します。

なお、配列(リストなど)の要素数がわからない場合は、len() 関数(len(配列名))で確認し、範囲内のインデックスとなるように修正します。

- 指定したインデックスは、配列の要素数の範囲内になっていますか？

### (エラー例)

```
File "c: /Users/user/Documents/test25.py", line 4, in <module>
    print('2020 年 4 月は、', days[4], '日あります')
IndexError: list index out of range
```

## ZeroDivisionError

0 による割り算がおこなわれたことによるエラー

(Zero とは「0」を、Division とは「割り算」を意味する英語)

### (対処方法)

違反が起きている行から、0 による割り算をしている式をみつけます。

式の入力ミスであれば修正し、除数(割る数)が 0 になることがある式ならば、if 文により除数(割る数)が 0 になったときの処理を付け加えます。

- 0 による割り算(除算/)をしていませんか？
- 0 による割り算(整数除算//、剰余演算%)をしていませんか？
- 除数(割る数)が 0 になることがある式ですか？(次のような if 文を追加しましょう)

```
if students == 0:  
    ave = 0  
else:  
    ave = goukei / students
```

### (エラー例)

```
File "c:/Users/user/Documents/test28.py", line 10, in <module>  
    print(sum(tokuten) / students)  
ZeroDivisionError: division by zero
```

(参考) Python » (Japanese 3.7.8) Documentation

<https://docs.python.org/ja/3.7/contents.html>

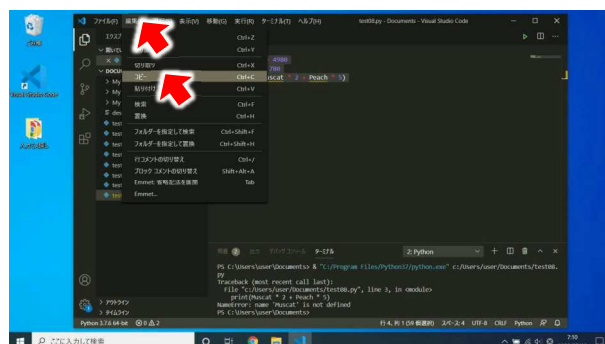
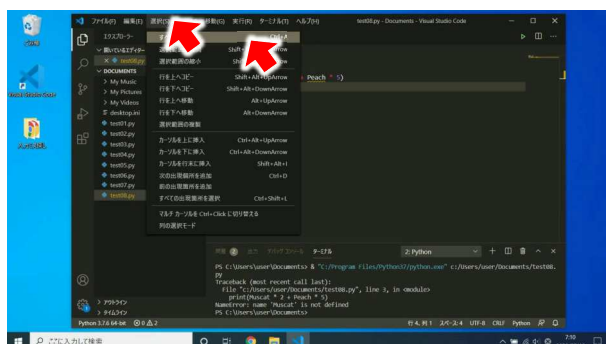
## 2 プログラム・ミスの箇所が見つからない場合の対処方法

この教材では、例題やチャレンジ問題を含む29のプログラム見本とあなたが入力したプログラムとの比較ができる仕組み「**入力ミス探し**」を用意しています。エラーが起きているものの、ミスの箇所が見つからない場合、「入力ミス探し」を利用してみてください。

(要インターネット接続)

### ①エディターVisual Studio Code でプログラムをコピー

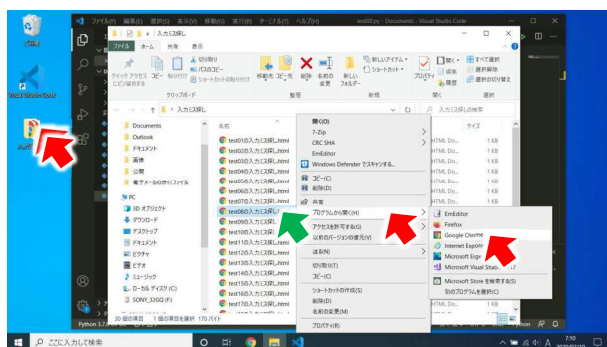
エディターVisual Studio Code で編集中のプログラムをコピーするため、**選択 > すべて選択** をクリックし、**編集 > コピー** をクリックします。



### ②Webブラウザ(Google Chrome)で「入力ミス探し」のショートカットを開く

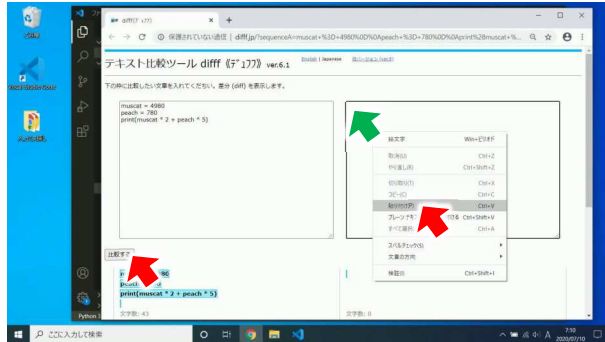
デスクトップにある「**入力ミス探し**」のフォルダーをダブルクリックし、比較したいプログラム名の書かれたファイルを**右クリック**、**プログラムから開く > Google Chrome** をクリックしてください(Webサイト diff が表示されます)。

(例: 例題08 test08.pyの入力ミス探しを使う場合)



### ③Webサイト diff でのテキスト比較

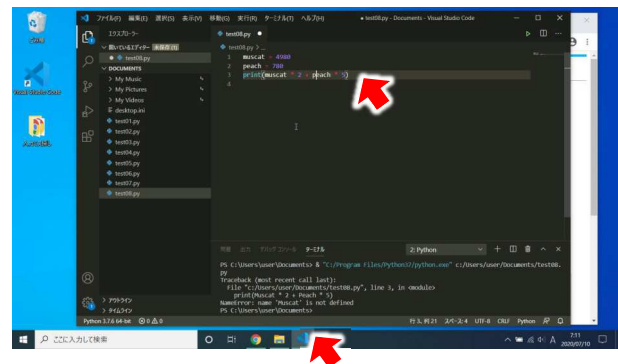
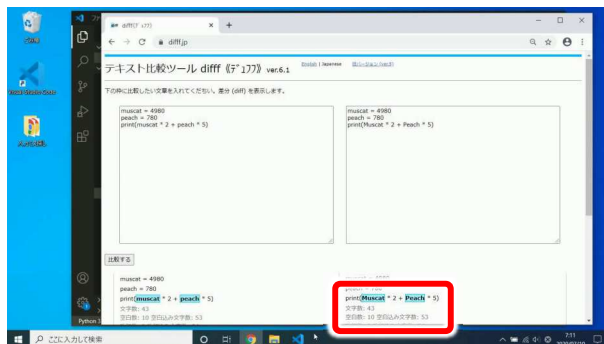
表示された Web サイト diff の左の欄には、プログラム見本が入力されているかと思えます。右の欄で**右クリック**、**貼り付け** をクリックし、**[比較する]** ボタンをクリックしてください(画面下に比較結果が表示されます)。



### ④エディター Visual Studio Code でプログラムを修正

Web サイト diff の画面下に現れる比較結果のうち、水色のついた箇所がプログラム見本と異なる場所です。

エディター Visual Studio Code に**切り替え**、見本と同じになるようにプログラムを修正してください。



なお、入力ミス探しの操作方法の流れは、ビデオで確認できます。

Win\_norm08

