Variable Assignments

PIERIAN DATA

Complete Python 3 Bootcamp

- We just saw how to work with numbers, but what do these numbers represent?
- It would be nice to assign these data types a variable name to easily reference them later on in our code!
- For example:
  - `my_dogs = 2`

Complete Python 3 Bootcamp

Rules for variable names

- Names can not start with a number.
- There can be no spaces in the name, use _ instead.
- Can't use any of these symbols :'",<>/?|\()!@#$%^&*~-+

Complete Python 3 Bootcamp

- Rules for variable names
  - It's considered best practice (PEP8) that names are lowercase.
  - Avoid using words that have special meaning in Python like "list" and "str"

Complete Python 3 Bootcamp

- Python uses **Dynamic Typing**
- This means you can reassign variables to different data types.
- This makes Python very flexible in assigning data types, this is different than other languages that are **"Statically-Typed"**

Complete Python 3 Bootcamp

```
my_dogs = 2
```

```
my_dogs = [ “Sammy” , “Frankie” ]
```

This is okay in Python!

Complete Python 3 Bootcamp

my_dogs = 2

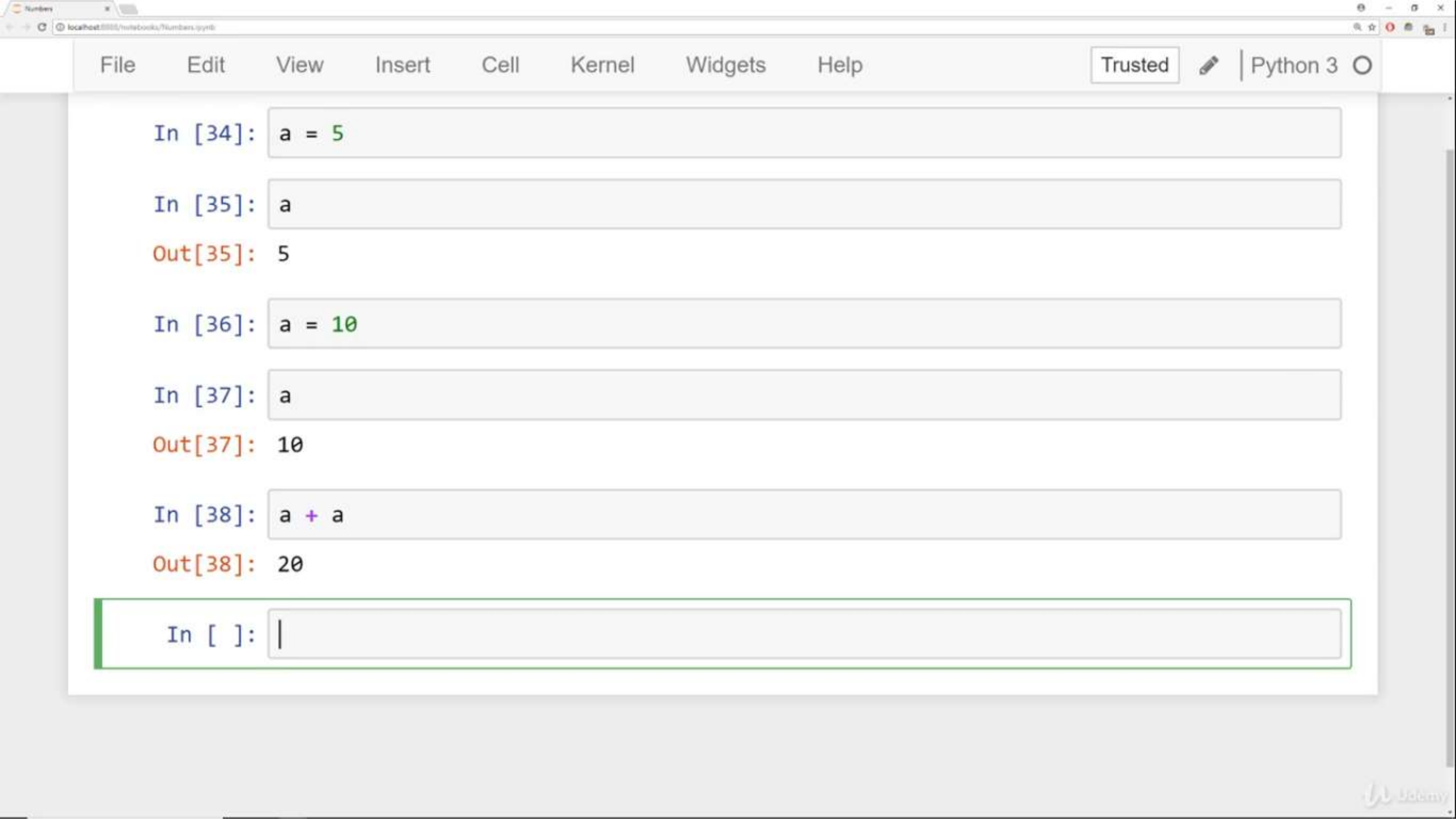
my_dogs = [“Sammy” , “Frankie”]

ERROR
in other
Languages!

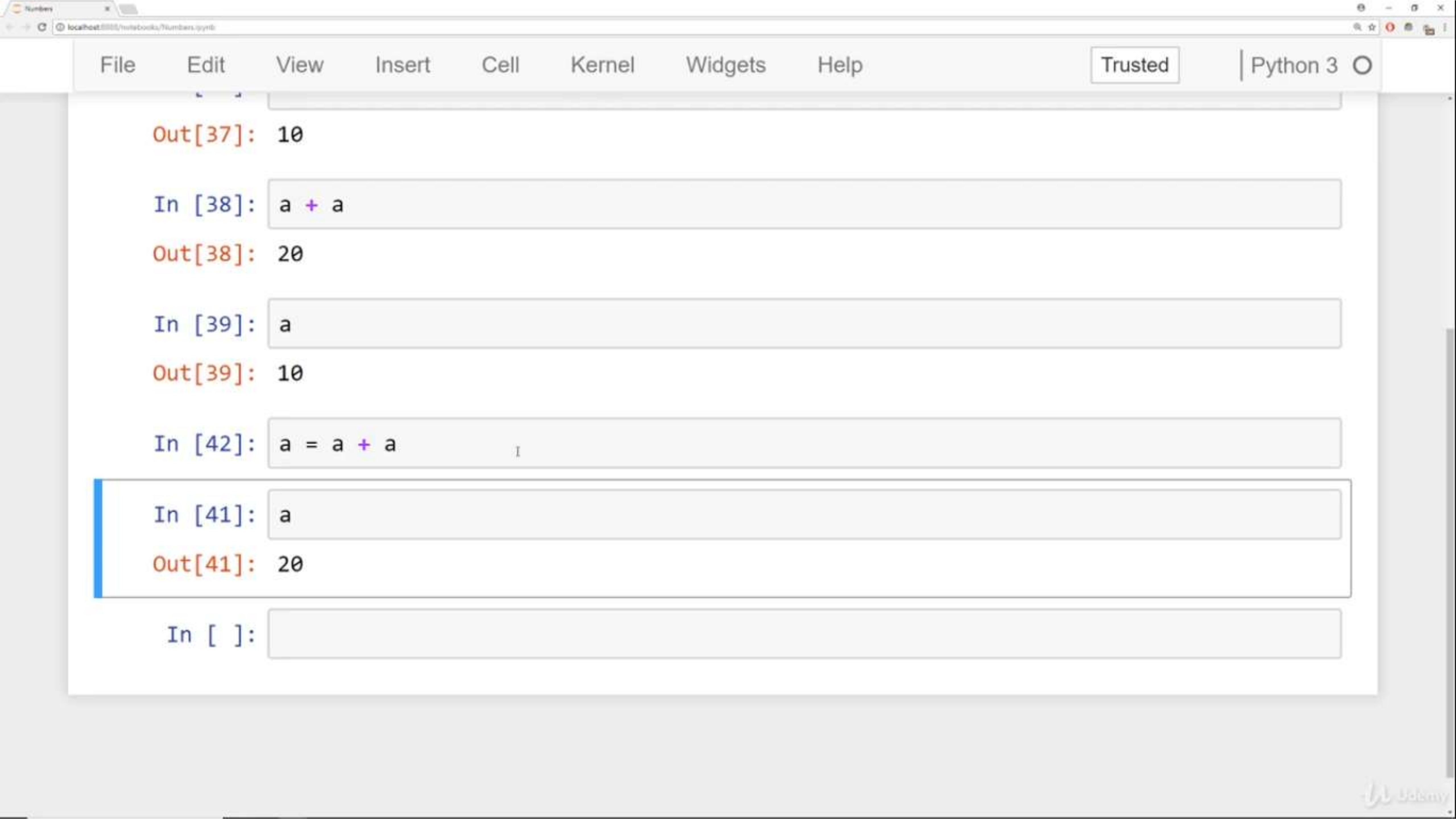
```
int my_dog = 1;

my_dog = “Sammy” ;  //RESULTS IN ERROR
```

Example of Static Typing
(C++)

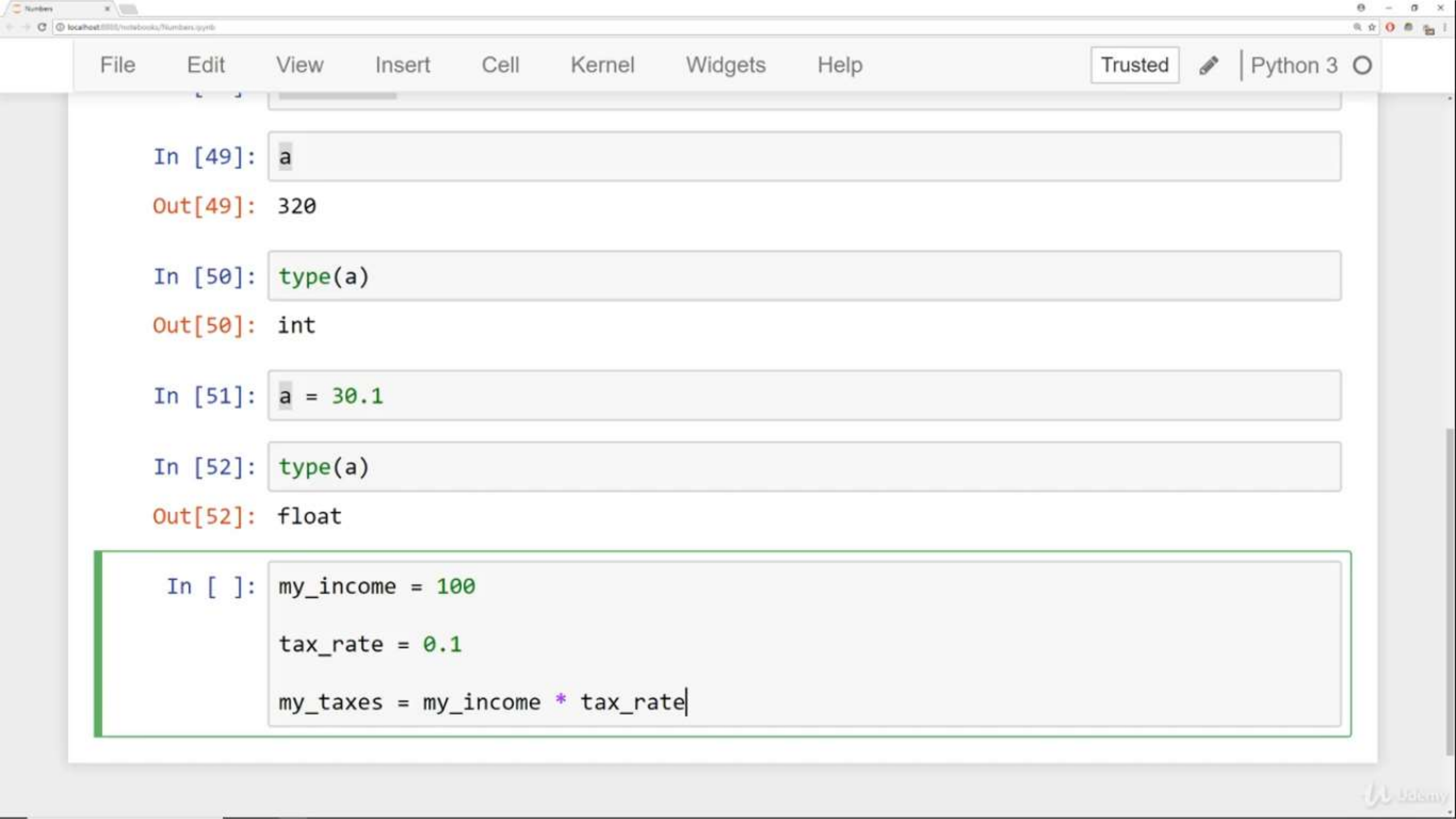
- Pros of Dynamic Typing:
  - Very easy to work with
  - Faster development time
- Cons of Dynamic Typing:
  - May result in bugs for unexpected data types!
  - You need to be aware of **type()**



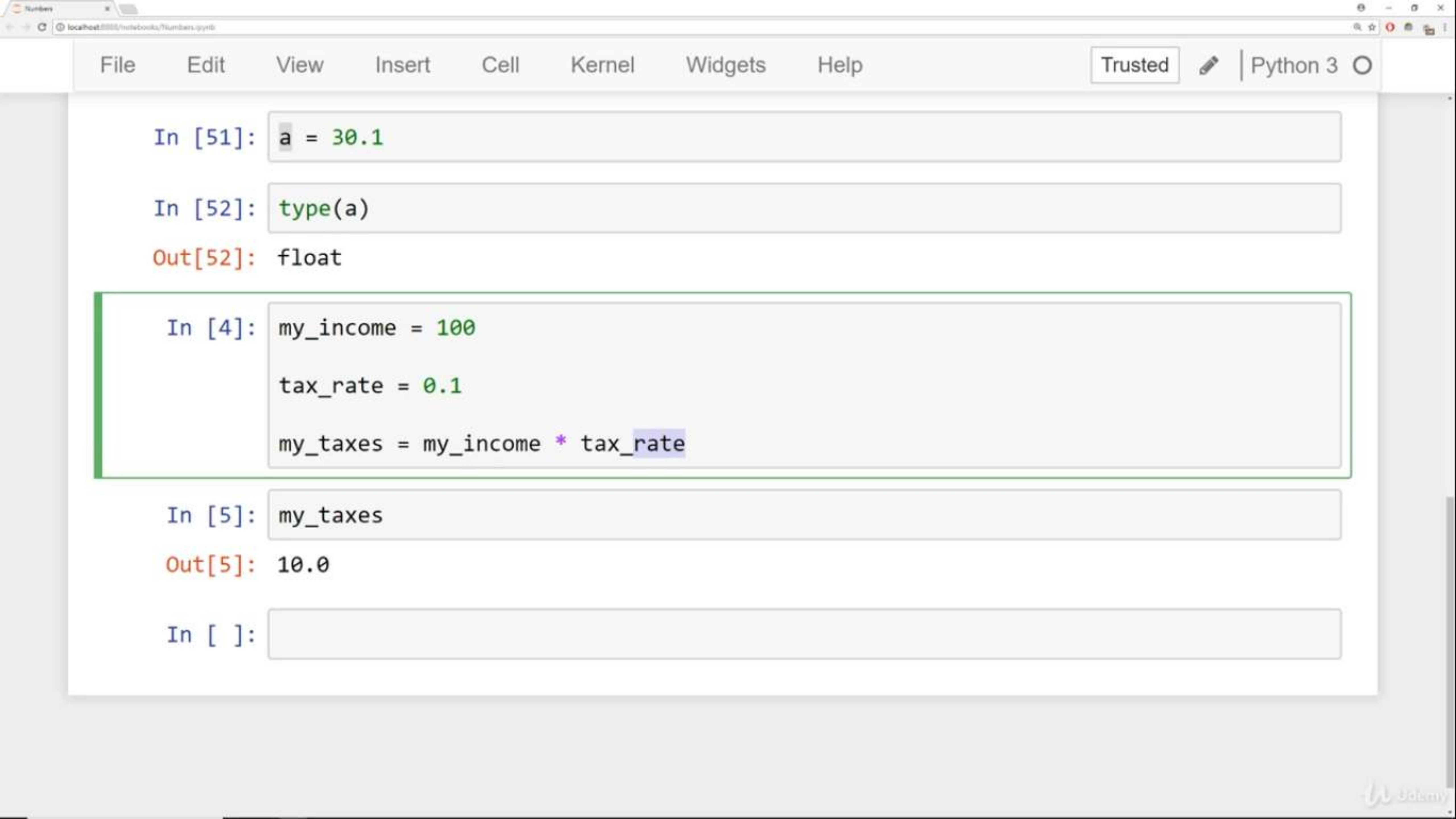
```
In [34]: a = 5

In [35]: a

Out[35]: 5

In [36]: a = 10

In [37]: a

Out[37]: 10

In [38]: a + a

Out[38]: 20

In [ ]:
```



```
Out[37]: 10

In [38]: a + a

Out[38]: 20

In [39]: a

Out[39]: 10

In [42]: a = a + a

In [41]: a

Out[41]: 20

In [ ]:
```



```
In [49]: a
Out[49]: 320

In [50]: type(a)
Out[50]: int

In [51]: a = 30.1

In [52]: type(a)
Out[52]: float

In [ ]: my_income = 100

        tax_rate = 0.1

        my_taxes = my_income * tax_rate
```



```
In [51]: a = 30.1

In [52]: type(a)

Out[52]: float

In [4]: my_income = 100

        tax_rate = 0.1

        my_taxes = my_income * tax_rate

In [5]: my_taxes

Out[5]: 10.0

In [ ]:
```