

DELETING VARIABLES

Variable
Assignment

Overwriting &
Deleting

Naming
Conventions

Tracking
Variables

The **del** keyword will permanently remove variables and other objects

```
price = 5
del price
print(price)
```

```
-----
NameError
Traceback (most recent call last)
/var/folders/f8/075hbnj13wb0f9yzh9k4ny
z00000gn/T/ipykernel_36120/432900761.p
y in <module>
      2 del price
      3
----> 4 print(price)
```

```
NameError: name 'price' is not defined
```

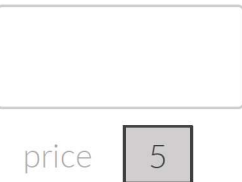
Python stores the value of 5 in memory when it is assigned to **price**

Then **del** removes it from memory

When we try to print **price**, we get an **Error**, as the variable no longer exists

NameError occurs when we reference a variable or object name that isn't defined

Memory



PRO TIP: Deleting objects is generally unnecessary, and mostly used for large objects (like datasets with 10k+ rows); in most cases, reassign the variables instead and Python will get rid of the old value!