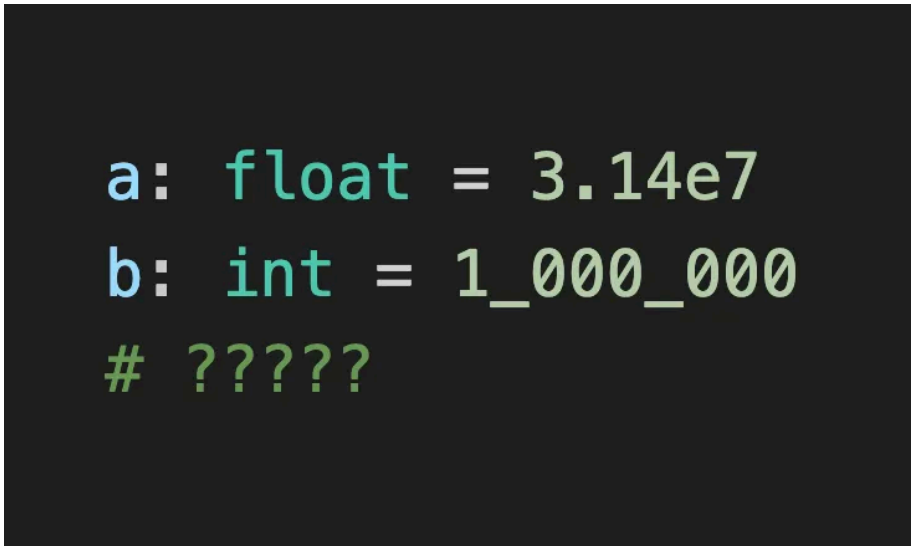
Member-only story

6 Things I Wish I Knew Earlier About Python Numbers

Liu Zuo Lin · Following
Published in Level Up Coding · 4 min read · Jun 26, 2024



1) We can write numbers using Scientific Notation

In science class, we could write 125000 as 1.25 x 10^5. We can too in Python

```
x: float = 1.25e5

print(x) # 125000.0
```

```
y: float = 3.45e-4

print(y) # 0.000345
```

- `1.25e5` means 1.25 * 10 ^ 5 (which is 100000)

- `3.45e-4` means 3.45 * 10 ^ -4 (which is 0.0001)

With this, we can easily type out very large and very small numbers without having to ensure that we've typed the correct number of zeros.

2) We can insert underscores in our numbers

so that we can *group* zeros together to make our number easier to read

```
x: int = 1_000_000_000

print(x) # 1000000000
```

^ here, we put one `_` between every 3 zeros so that we can tell at first glance that this is 1 billion.

```
y: int = 1_0_0_00_000

print(y) # 10000000
```

^ note that it is *not compulsory* to have 3 zeros per group. We can have as many zeros per group as we want, but I just stick with 3 because it's the easiest to read.

3) We can round() to negative decimal places

when we `round()` a number to -2 decimal places, we round our number to the nearest 100.

```
x: int = 123456

print(round(x, -2)) # 123500
```

when we `round()` a number to -3 decimal places, we round our number to the nearest 1000.

```
x: int = 123456

print(round(x, -3)) # 123000
```

when we `round()` a number to -4 decimal places, we round our number to the nearest 10000.

```
x: int = 123456

print(round(x, -4)) # 120000
```

4) We can use divmod() to perform // and % concurrently

```
a, b = divmod(57, 10)

print(a) # 5
print(b) # 7
```

The built-in `divmod(a, b)` function takes in 2 integers `a` and `b`, and returns a tuple containing `(quotient, remainder)`

- 57 divided by 10 is 5 (quotient), with a remainder of 7
- when we do `a // b`, we get the *quotient*
- when we do `a % b`, we get the *remainder*

The `divmod` function allows us to do this concurrently, thus saving us one line of code whenever we need to do `//` and `%` at the same time.

5) Positive and negative infinity

When we do `float('inf')`, we actually get positive infinity.

```
x: float = float('inf')

print(x) # inf
```

And when we do `float('-inf')`, we get negative infinity.

```
y: float = float('-inf')

print(y) # -inf
```

Positive infinity is larger than all other numbers, while negative infinity is smalle rthan all other numbers.

```
x: float = float('inf')

print(x > 1)          # True
print(x > 1_000_000)  # True
print(x > 3.14e100)   # True
```

```
y: float = float('-inf')

print(y < -1)          # True
print(y < -1_000_000)  # True
print(y < -1e1000)     # True
```

We can use positive of negative infinity if we ever need a number that is larger/smaller than all other numbers

6) The built-in decimal module

Python has a built-in `decimal` module that we can simply import, and it provides functionality to deal with decimal numbers

```
from decimal import Decimal

a: Decimal = Decimal(1) / Decimal(3)

print(a) # 0.33333333333333333333333333333
```

But why would we use this over normal `float` numbers?

- `float` numbers are approximate
- `float` numbers might have floating point inaccuracy
- `Decimals` are designed to be more precise

```
a: float = 2.0
b: float = 2.000000000000000000000001

print(a == b) # True
```

^ above a certain point, floating point inaccuracy kicks in, which might be problematic for tiny numbers.

7) (bonus) Complex numbers

Remember in math class where we had to deal with this weird concept known as complex numbers? We can do this in Python too.

```
a: complex = complex(5, 7)

print(a)        # (5+7j)
print(type(a))  # <class 'complex'>
```

^ here, `5 + 7j` means *5 + 7i (real 5 + imaginary 7)*

```
a: complex = complex(5, 7)

print(a.real)  # 5.0
print(a.imag)  # 7.0
```

^ we can use the `.real` and `.imag` to extract the real and imaginary components of our imaginary number.

```
import cmath
```

^ we can import the built-in `cmath` module to perform various complex number operations (which I won't go into detail here)

I've never used this at work before (why would I lol), but this was an interesting fact to know anyway.

Complimentary Python Joke

> What's the scariest data type?
>
> BOO-lean

If You Wish To Support Me As A Creator

- Buy my book! — **101 Things I Never Knew About Python**
- Where to find it: https://payhip.com/b/vywcf
- *Clap 50 times for this story*
- *Leave a comment telling me your thoughts*

- *Highlight your favourite part of the story*

Thank you! These tiny actions go a long way, and I really appreciate it!

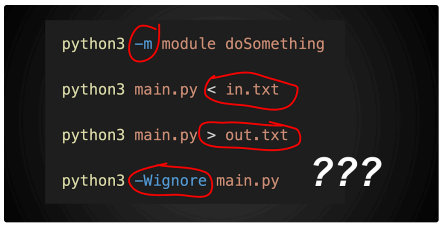
YouTube: https://www.youtube.com/@zlliu246

LinkedIn: https://www.linkedin.com/in/zlliu/

Written by Liu Zuo Lin

96K Followers · Writer for Level Up Coding

My Substack Newsletter: https://zlliu.substack.com | 101 Things I Never Knew About Python: https://payhip.com/b/vywcf



Liu Zuo Lin in Level Up Coding

13 Python Command Line Things I Regret Not Knowing Earlier

Read free...

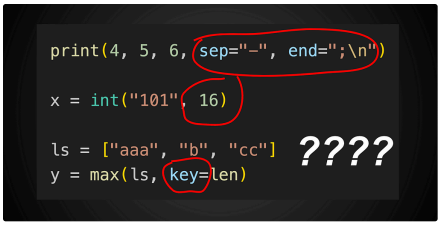
Oct 19 667 12

Bryson Meiling in Level Up Coding

Stop making your python projects like it was 15 years ago...

I have a few things I've seen across companies and projects that I've seen...

Sep 28 4.3K 45



Hayk Simonyan in Level Up Coding

STOP using Docker Desktop: Faster Alternative Nobody Uses

Ditch Docker Desktop and try this faster, lighter tool that will make your life easier!

Oct 8 2.3K 41

Liu Zuo Lin in Level Up Coding

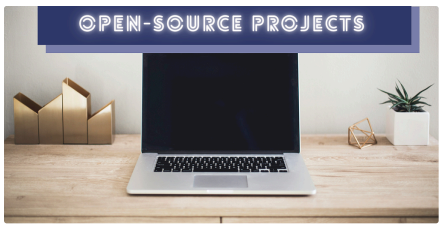
12 Python Built-in Function Things I Regret Not Knowing Earlier

Friend Link Here

Oct 12 873 5

See all from Liu Zuo Lin | See all from Level Up Coding

Recommended from Medium



Shalitha Suranga in Level Up Coding

Abdur Rahman in Stackademic

Impressive Open-Source Projects That Only A Few Programmers...

Priceless open-source innovations that impress any programmer

Oct 21 976 11

20 Python Scripts To Automate Your Daily Tasks

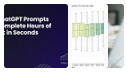
A must-have collection for every developer

Oct 6 1.2K 9

Lists

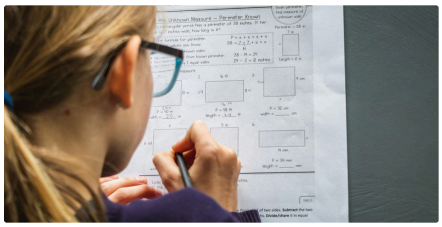
Coding & Development
11 stories · 878 saves

General Coding Knowledge
20 stories · 1688 saves



ChatGPT prompts
50 stories · 2157 saves

ChatGPT
21 stories · 853 saves

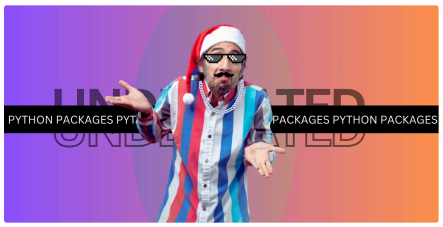


Egor Howell in Towards Data Science

4 Years of Data Science in 8 Minutes

What I have learned in my 4+ year journey of studying data science

5d ago 825 13



Abhay Parashar in The Pythoneers

23 Game-Changing Python Packages You Are Missing Out On

Make Your Life Easy By Exploring These Hidden Gems

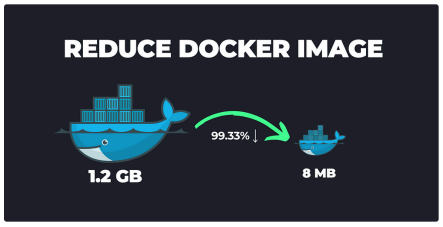
Oct 21 1.3K 9

Ignacio de Gregorio

Apple Speaks the Truth About AI. It's Not Good.

Are We Being Lied To?

6d ago 3.7K 119



Dipanshu in AWS in Plain English

Docker pros are shrinking images by 99%: The hidden techniques yo...

Unlock the secrets to lightning-fast deployments and slashed costs—before yo...

Sep 18 2.4K 11

See more recommendations