

Lecture 3, 13 August 2024

Madhavan Mukund

<https://www.cmi.ac.in/~madhavan>

Programming and Data Structures with Python

Prime numbers

Is n a prime?

Find factors — how to?

All previous numbers $1..n-1$

⇒ Only 1 factor means prime

⇒ 1 is a prime?

Better to check $2..n-1$

0 factors ⇒ prime

How to generate $2..n-1$?

List $[2, 3, \dots, n-1]$

"like lists" $d.keys()$ for dictionary d

$range(n)$

More generally $range(m, n)$ $m, m+1, \dots, n-1$

← Assume prime

for i in range(2, n):

if i divides n , not a prime ← Set status to non prime

Data types - types of values

numbers

text

boolean

True / False

isprime = True

for i in range(2, n):

2, 3, ..., n-1

if i divides n: ← ?

isprime = False



isprime reflects status of primality of n

Numbers

In Math

Integers \subseteq Reals

$$\mathbb{Z} \subseteq \mathbb{R}$$

Arithmetic

$+, -, *, /$ division

$$3/7 = ?$$



int - integer

float - floating point

Prog langs store integers & reals differently



$$\begin{array}{c} \bullet 602 \times 10^{24} \\ \uparrow \end{array}$$

$$\text{not } 6.02 \times 10^{23}$$

int }
float } +, -, *, / \hookrightarrow always produces a float

$$7 \div 3 \Rightarrow \text{quotient } 2, \text{ remainder } 1$$

$n \% i$

$n \bmod i$

remainder

$n // i$

quotient

isprime = True

for i in range(2, n):

if n % i == 0:

isprime = False

Stop at \sqrt{n}

import from math
library

Generate all primes upto n

for i in range $(2, n+1)$:

if i is a prime, record it - use a list

```
primelist = []
```

```
for i in range(2, n+1):
```

```
    if i is prime: ← reuse old code
```

```
        primelist = primelist + [i]
```

↙ makes i
into a list

```
        primelist.append(i)
```

Modularize code into independent units - functions

```
if isprime(i):
```

≡

```
def isprime(n):
```

```
    status = True
```

```
    for i in range(2, n):
```

```
        if n%i == 0:
```

```
            status = False
```

```
    return(status)
```