

What is Programming and Python?

Programming: writing codes that allow you to create your own set of instructions for your device to carry out.

Python: a language we can use for programming



I'm a biology major, why do I need it?

It allows us to work with large data sizes faster, more efficiently and more precisely



Variables

Variables are used to store information or values such as a number or words. To create a variable in Python, write the name of the variables and assign it to a value.

```
variable_name = value
```

Naming variables

1. Must begin with a letter (a - z, A - B) or underscore (_)
2. Other characters can be letters, numbers or _
3. Do not use Python keywords
4. Case Sensitive
5. Meaningful names

Lists

- Lists are a collection of data which can be of different data types.
- Lists are mutable, meaning you can add, delete or change the elements in the list.

Slicing lists

- Values can be sliced from lists by specifying its position
- When extracting a range of items, use [Start:End:Step]

Dictionary

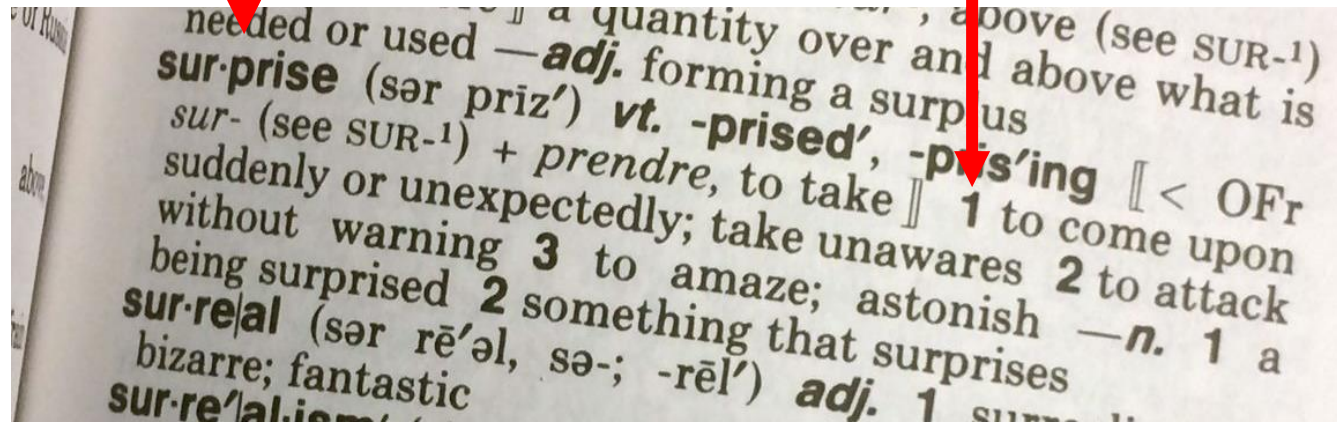
- Dictionary is similar to a list, but the order of items doesn't matter, and they aren't selected by an offset such as 0 or 1. Instead, you specify a unique key to associate with each value.

`dictionary = { key : item }`

Dictionary

Key

Value



For loops

Python makes frequent use of iterators, for good reason. They make it possible for you to traverse data structures without knowing how large they are or how they are implemented.

It is possible to loop through:

- Strings
- Integers
- Lists
- Etc

For loops

for each item in this list:
do this

For loops

```
prime_numbers = [2, 3, 5, 7, 11]
```

```
for each number in the list of prime_numbers:  
    multiply by two  
    print each number
```

$$2 \times 2 = 4$$

$$3 \times 2 = 6$$

$$5 \times 2 = 10$$

$$7 \times 2 = 14$$

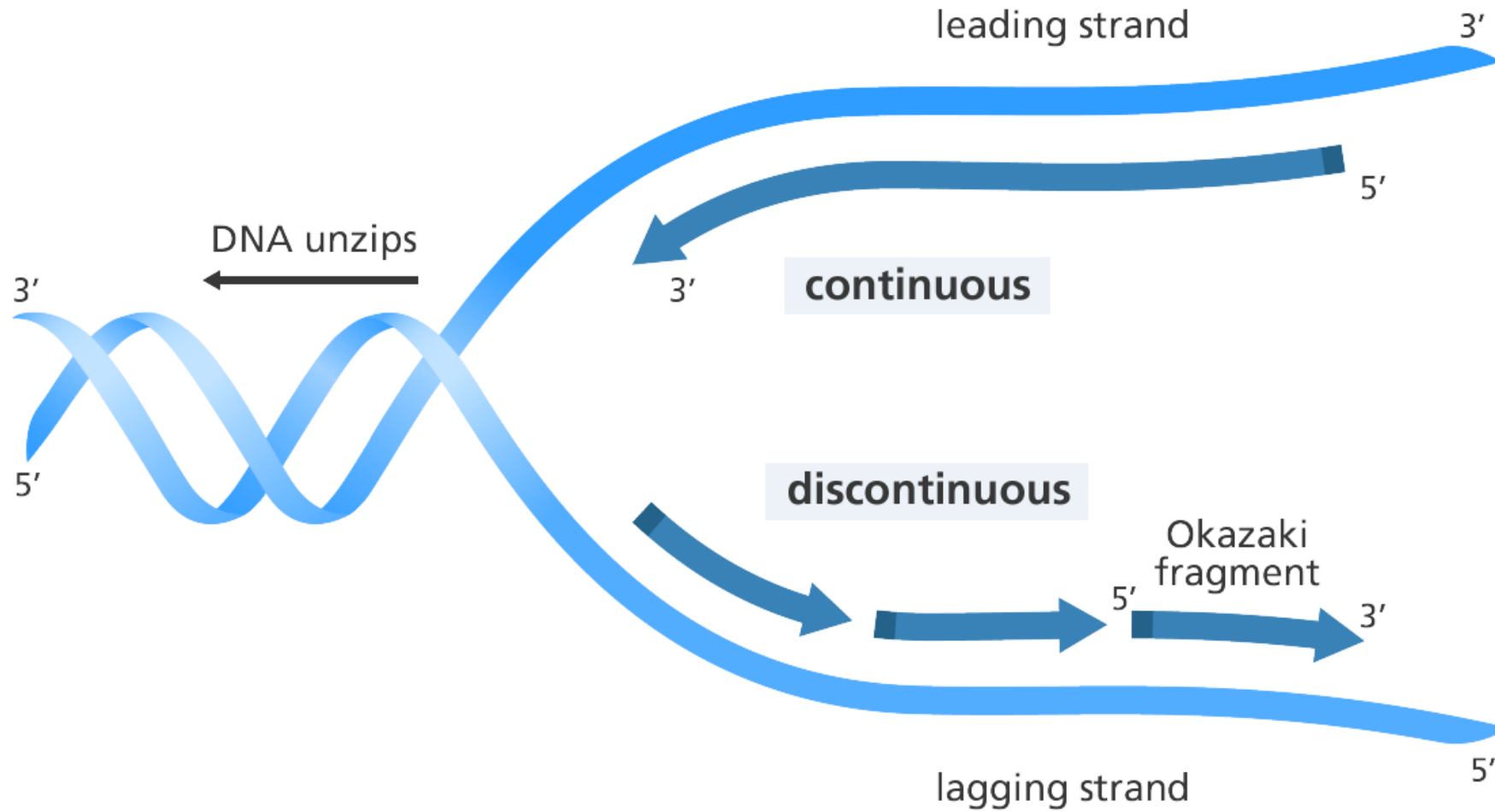
$$11 \times 2 = 22$$

Outcome: 4 6 10 14 22



Complementary Strand of DNA

DNA replication fork



Given a strand of DNA, how do we find its complementary strand?

3' A C G T A C G T G A C G 5'

1. Identify complement bases

A → T

C → G

G → C

T → A

2. Replace the bases, one at a time using the complement bases



2. Replace the bases, one at a time using the complement bases



2. Replace the bases, one at a time using the complement bases



2. Replace the bases, one at a time using the complement bases

3' A C G T A C G T G A C G 5'

5' T G C A T G C A C T G C 3'

Solution

Making a dictionary

```
complement = {  
    'A': 'T',  
    'C': 'G',  
    'G': 'C',  
    'T': 'A'  
}
```

Given a strand of DNA, how do we find its complementary strand?

complement = {

'A': 'T',

'C': 'G',

'G': 'C',

'T': 'A'

}



5' A C G T A C G T G A C G 3'

3' T 5'

Given a strand of DNA, how do we find its complementary strand?

complement = {

'A': 'T',

'C': 'G',

'G': 'C',

'T': 'A'

}



5' A C G T A C G T G A C G 3'

3' T G

5'

Given a strand of DNA, how do we find its complementary strand?

```
complement = {  
    'A': 'T',  
    'C': 'G',  
    'G': 'C',  
    'T': 'A'  
}
```

5' A C G T A C G T G A C G 3'
3' T G C A T G C A C T G C 5'

Remember to flip the strand!

5' C G T C A C G T A C G T 3'

```
dna = 'GCGCGCGCGCGAAATTTT'  
base_dict = {'A': 'T', 'C': 'G', 'G': 'C', 'T': 'A'}  
complement = ''  
for nucleotide in dna:  
    complement += base_dict[nucleotide]  
reverse_complement = complement[::-1]  
print(reverse_complement)
```