

Welcome to Python!

March 29, 2022





Case Study: Hello World

// C++

```
$ g++ helloWorld.cpp  
$ ./a.out
```


Case Study: Int Size

```
// C++  
#include <iostream>  
using namespace
```

```
$ g++ intSize.cpp  
$ ./a.out  
4
```



```
>>> import this
```

The Zen of Python, by Tim Peters

```
>>> import this
```

In the face of ambiguity, refuse the temptation to guess.

There should be one-- and preferably only one --obvious way to do it.

Although that way may not be obvious at first unless you're Dutch.

Now is better than never.

Although never is often better than **right** now.

If the implementation is hard to explain, it's a bad idea.

If the implementation is easy to explain, it may be a good idea.

Namespaces are one honking great idea -- let's do more of those!

Learning Goals

After CS 41, students will be able to...

1. Compare and contrast Python's language design with other languages they've seen before.
- 2.



Today's Agenda





TT 167l 167lt 167love

Something to get used to is no explicit distinction of
types by the user



Now this does not mean that there are no types in Python, they are just hidden

n a m e = " T a r a

Lists



Questions?

Control Flow

if, else





.try Except

- For trying dangerous code

Continue

```
#make every number odd  
my_list = [1,2,3,4,5]  
  
for elem in my_list:  
    if elem % 2 == 0:  
        elem+=1  
    else:
```

While Loops

```
while true:  
    number = int(input("Give a number: " ))  
    if number > 100:  
        break
```


Thank you!