Addition of two numbers.

```
print(18+27)
```

```
45
```

```
a=19
b=27
c=a+b
print(c)
```

```
46
```

```
a=int(input("Enter first number:")) # Enter first number:
b=int(input("Enter second number:"))
c=a+b
print("The sum is",c)
```

```
Enter first number:17
Enter second number:28
The sum is 45
```

+ Code + Text

Built-in types:

1. Makes programs easy to write.
2. Easy to extend.
3. More efficient than custoem data structures.
4. Are a standard part of the language.

Table 4-1. Built-in objects preview

Object type	Example literals/creation
Numbers	`1234, 3.1415, 3+4j, Decimal, Fraction`
Strings	`'spam', "guido's", b'a\x01c'`
Lists	`[1, [2, 'three'], 4]`
Dictionaries	`{'food': 'spam', 'taste': 'yum'}`
Tuples	`(1, 'spam', 4, 'U')`
Files	`myfile = open('eggs', 'r')`
Sets	`set('abc'), {'a', 'b', 'c'}`
Other core types	Booleans, types, None
Program unit types	Functions, modules, classes (Part IV, Part V, Part VI)
Implementation-related types	Compiled code, stack tracebacks (Part IV, Part VII)

1. Numbers

```
print( 2 ** 100) #2 to the power of 100
len(str(2 ** 100))
```

```
1267650600228229401496703205376
31
```

```
print(100 ** 2)
```

```
10000
```

```
print( 3.1415 * 2)
```

```
6.283
```

```
import math  #additinal python module
pi = 3.141
print(pi)
print(math.pi)
```

```
3.141
3.141592653589793
```

```
print(math.sqrt(250))
```

```
15.811388300841896
```

```
print((250)**(1/2))
```

```
15.811388300841896
```

```
import random
random.choice([2,4,7,10]) # Return a random number from the list.
```

```
7
```

```
random.random() # Return random number between 0.0 and 1.0
```

```
0.17689487646977897
```

2. Strings

```
 S = 'Spam'
```

```
S
```

```
 S = "Spam"
```

```
print(S)
```

```
print(len(S))
```

```
S[0]
```

```
S[3]
```

```
S[-2] #backward indexing
```

```
S[1:]  # Slicing
```

```
S[0:3]
```

```
S[:3]
```