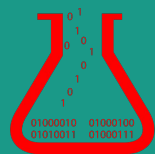


I/O



---

## Formatting Output

# Text output | The Basics

## Store a string in a variable:

`s='the black cat crossed the road backwards'`

## Simple concatenation:

`>>>print ('this is my text: ' + s)`

*this is my text: the black cat crossed the road*

## Change the newline end default:

`>>> print ('this is my text: ' + s, end="***")`

**this is my text: the black cat crossed the road backwards\*\*\*>>>**

# Text output | The Basics

## Multiply:

```
s='the black cat crossed the road backwards'
```

## Simple concatenation:

```
>>>print ('this is my text: ' + s)
```

```
this is my text: the black cat crossed the road
```

## Change the newline end default:

```
>>> print ('this is my text: ' + s, end="***")
```

```
this is my text: the black cat crossed the road backwards***>>>
```

# Text output | The Basics

## The old s(%) style

```
>>> print ('this is my text: %10s ' % (s))
```

```
>>> print ('this is my text: %s ' % (s))  
this is my text: (有)word  
>>> print ('this is my text: %10s ' % (s))  
this is my text:      (有)word
```

## Text output |

### The new {} style


```
print ("This is my text: {sentence:<20s}".format(sentence=s))  
print ("This is my text: {sentence}".format(sentence=s))  
print ("This is my text: {}".format(s))
```

```
>>> print ("This is my text: {sentence:<20s}".format(sentence=s))  
This is my text: (有)word  
>>> print ("This is my text: {}".format(s))  
This is my text: (有)word
```

---


## Working with Files

# The basic pattern | Python Functions

- 
1. `.open(file, mode)`,
    - a. **open modes** (`'wt'`, `'rt'`, `'at'`)
  2. **Action on file:**
    - a. `write()`, `read()`, `readlines()`, `readline()`
  3. `.close()`



# Loading files | Python Functions

- 
1. `.open(file, mode), .close()`
  2. open modes ('w', 'r', 'a')
  3. WITH-AS # you don't need to close this one
  4. `read()`
  5. `readlines()` # reads all lines as a list
  6. `readline()` #reads one line in at a time

---

## Working with Common Formats

# File Formats



**JSON** , CSV, XML

**packages**, json , csv, , xml.etree

json.load()

json.dumps()

csv.writer()

csv.reader()

writerows()

# Questions?

---



**Contact:**

**Denis Vrdoljak**  
**denis@bds.group**

---