

Python Fundamentals

Strings and Collections

Austin Bingham
🐦 @austin_bingham
austin@sixty-north.com



Presenter

Robert Smallshire
🐦 @robsmallshire
rob@sixty-north.com



pluralsight 
hardcore developer training

Collections



str

bytes

list

dict

for-loops



str

immutable sequences of Unicode codepoints

String Literals

`'This is a string'`

`"This is also a string"`

Moment of Zen

Practicality beats
purity

Beautiful text strings
Rendered in literal form
Simple elegance



Strings with Newlines

1. **Multiline** strings
2. **Escape** sequences



‘\r\n’

carriage-return newline



OSX

‘\n’

newline

Universal Newlines

‘\n’

PEP 278

<http://www.python.org/dev/peps/pep-0278/>

Escape Sequences

Sequence	Meaning
<code>\newline</code>	Backslash and newline ignored
<code>\\</code>	Backslash (\)
<code>\'</code>	Single quote (')
<code>\a</code>	ASCII Bell (BEL)
<code>\b</code>	ASCII Backspace (BS)
<code>\f</code>	ASCII Formfeed (FF)
<code>\n</code>	ASCII Linefeed (LF)
<code>\r</code>	ASCII Carriage Return (CR)
<code>\t</code>	ASCII Horizontal Tab (TAB)
<code>\v</code>	ASCII Vertical Tab (VT)
<code>\ooo</code>	Character with octal value ooo
<code>\xhh</code>	Character with hex value hh
Only recognized in string literals	
<code>\N{name}</code>	Character named name in the Unicode database
<code>\uxxxx</code>	Character with 16-bit hex value xxxx
<code>\Uxxxxxxxx</code>	Character with 32-bit hex value xxxxxxxx

Escape Sequences

Sequence	Meaning
<code>\newline</code>	Backslash and newline ignored
<code>\\</code>	Backslash (\)
<code>\'</code>	Single quote (')
<code>\a</code>	ASCII Bell (BEL)
<code>\b</code>	ASCII Backspace (BS)
<code>\f</code>	ASCII Formfeed (FF)
<code>\n</code>	ASCII Newline (LF)
<code>\r</code>	ASCII Carriage Return (CR)
<code>\t</code>	ASCII Horizontal Tab (HT)
<code>\v</code>	ASCII Vertical Tab (VT)
<code>\ooo</code>	Character with octal value ooo
<code>\xhh</code>	Character with hex value hh
Only recognized in string literals	
<code>\N{name}</code>	Character named name in the Unicode database
<code>\uxxxx</code>	Character with 16-bit hex value xxxx
<code>\Uxxxxxxxx</code>	Character with 32-bit hex value xxxxxxxx

http://docs.python.org/3/reference/lexical_analysis.html#strings

No Separate Character Type

‘a long string’
‘x’ } str

No separate character type
"characters" are simply *one element strings*

Python Strings Are Unicode

Source encoding: **UTF-8**
UTF-8 literals



bytes

immutable sequences of bytes

Bytes Literals

`b' data '`

`b" data "`

Converting Between Strings and Bytes

decode

str

bytes

encode

Encodings

<http://docs.python.org/3/library/codecs.html#standard-encodings>



list

mutable sequences of objects

List Literals

[a, b, c, d]



dict

mutable mappings of keys to values

Dict Literals

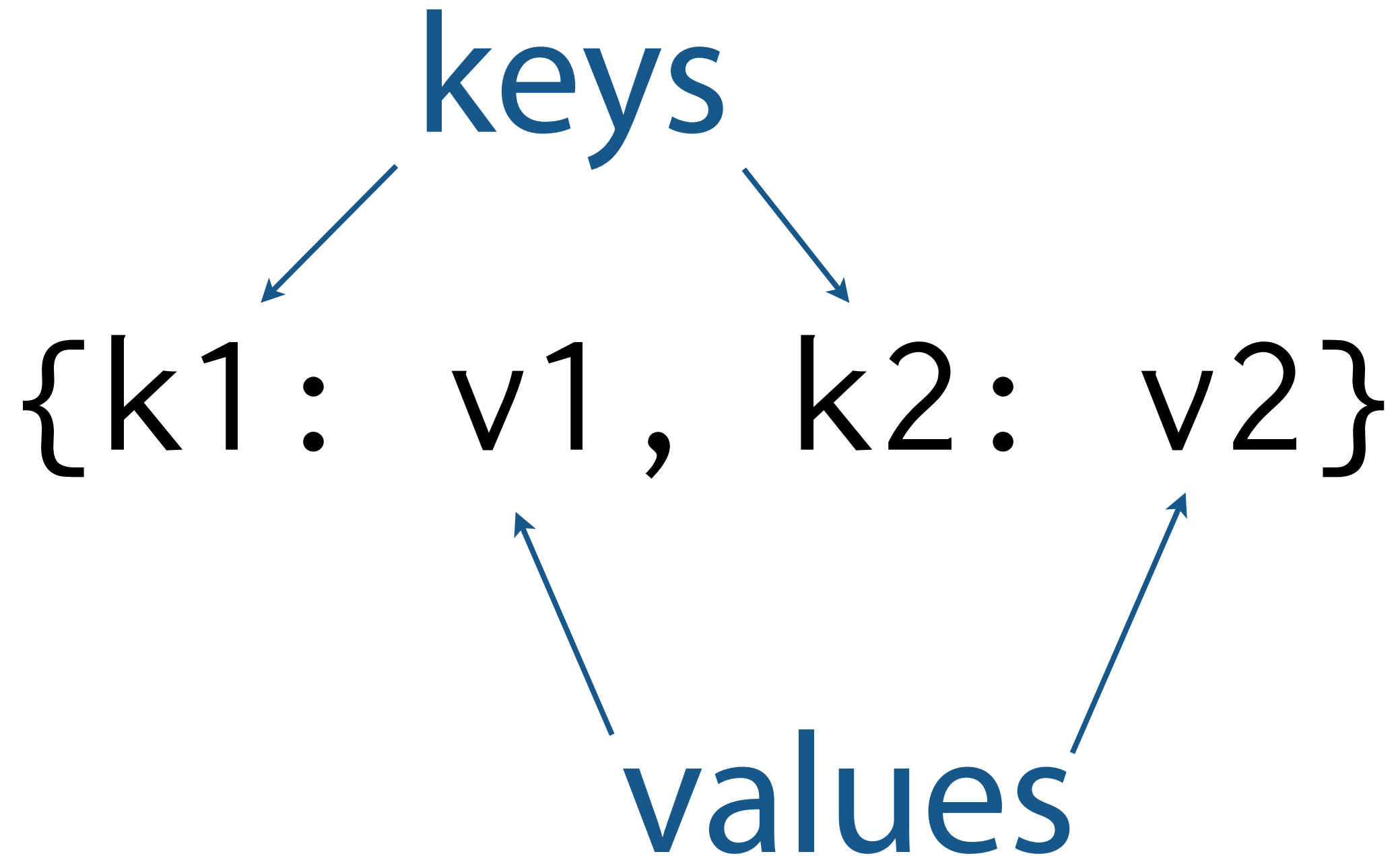
`{k1: v1, k2: v2}`

Dict Literals

keys

{k1: v1, k2: v2}

Dict Literals





for-loop

visit each item in an iterable series

For-Loop Syntax

```
for item in iterable:  
    ...body...
```

Exit the REPL



Ctrl-Z



OSX

Ctrl-D



Putting it all together

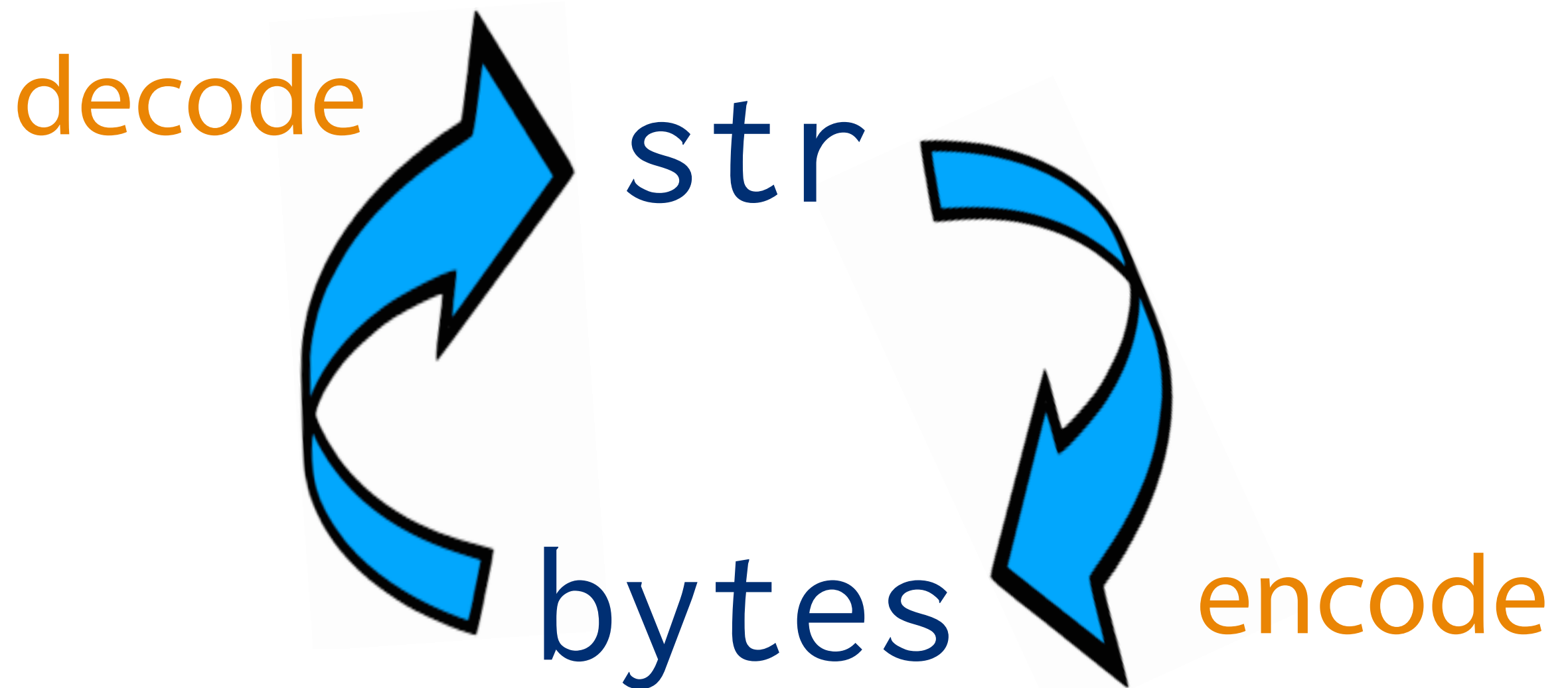
- `urllib.urlopen()`
- `with-statement`
- `list`
- `for-loop`
- `bytes.split()`

Recall: Bytes Literals

`b' data '`

`b" data "`

Recall: Converting Between Strings and Bytes





Putting it all together

- `urllib.urlopen()`
- `with-statement`
- `list`
- `for-loop`
- `bytes.split()`
- `bytes.decode()`
- `str.split()`



Summary: Strings and Bytes

- Single- and multi-line string quoting
- Adjacent string literal concatenation
- Universal newlines
- Escape sequences for control characters
- Raw strings suppress the escaping mechanism
- Convert other types with the `str()` constructor
- Zero-based square-bracket indexing of strings
- Rich variety of string methods
- Python 3 source encoding is UTF-8
- `bytes` is a sequence of bytes, `str` is a sequence of Unicode codepoints
- `bytes` literals prefixed with a lowercase `b`



Summary: Strings and Bytes

- Single- and multi-line string quoting
- Adjacent string literal concatenation
- Universal newlines
- Escape sequences for control characters
- Raw strings suppress the escaping mechanism
- Convert other types with the `str()` constructor
- Zero-based square-bracket indexing of strings
- Rich variety of string methods
- Python 3 source encoding is UTF-8
- `bytes` is a sequence of bytes, `str` is a sequence of Unicode codepoints
- `bytes` literals prefixed with a lowercase `b`
- Convert `str` to bytes with `encode()`, bytes to `str` with `decode()`



Summary: Lists

- Lists are mutable, heterogeneous sequences of objects
- List literals delimited by square brackets, items separated by commas
- Zero-based, square-bracket indexing to retrieve objects
- Square-bracket assignment to replace objects
- Grow lists with `append()`
- Construct from other sequences using `list()` constructor

[a, b, c, d]



Summary: Dictionaries

- Dictionaries associate keys with values
- Literal dicts delimited by curly braces
- Literal key-value pairs separated by commas, with a colon between each key and value

`{k1: v1, k2: v2}`



Summary: For-loops

- Take items one-by-one from an iterable object, binding a name to the current item
- Correspond to *for-each* loops in other languages

```
for item in iterable:  
    ...body...
```



- Tak
- curr
- Cor

Next time in Python Fundamentals

- &• **functions**
- &• **modules**
- &• **execution model**
- &• **command-line arguments**

f

