python™

str

immutable sequences of Unicode codepoints

String Literals

‘This is a string’

“This is also a string”

Strings with Newlines

1. **Multiline** strings

2. **Escape** sequences

Windows

‘\r\n’

carriage-return newline

Linux

OS X

‘\n’

newline

Universal Newlines

PEP 278

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

Escape Sequences

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

Escape Sequences

Sequence	Meaning
\newline	Backslash and newline ignored
\\	Backslash (\)
\’	Single quote (‘)
\a	ASCII Bell (BEL)
\b	ASCII Backspace (BS)
\f	ASCII Formfeed (FF
\n	
\r	
\t	
\v	
\ooo	Character with octal value ooo
\xhh	Character with hex value hh
Only recognized in string literals	
\N{name}	Character named name in the Unicode database
\uxxxx	Character with 16-bit hex value xxxx
\Uxxxxxxxx	Character with 32-bit hex value xxxxxxxx

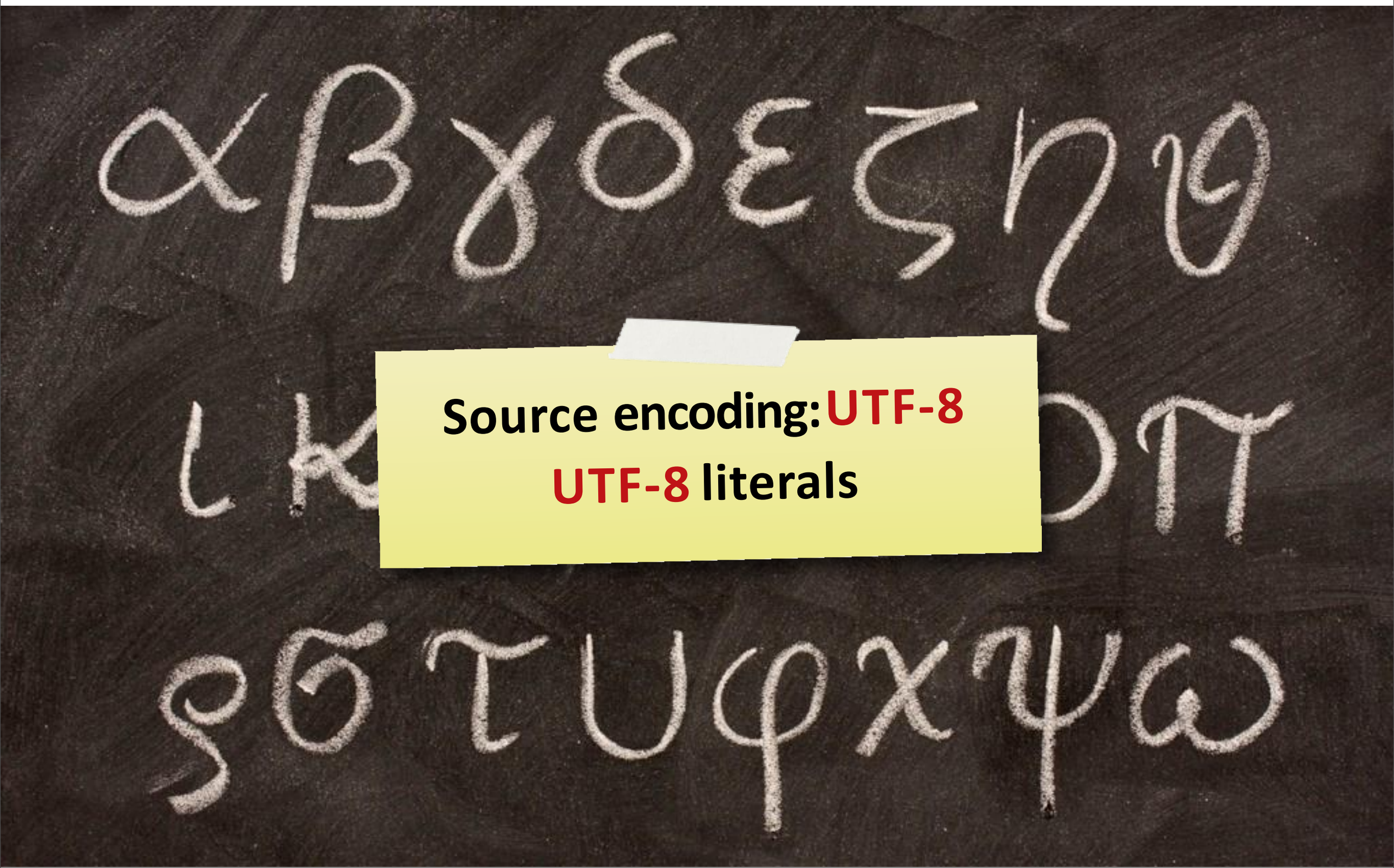
http://docs.python.org/3/reference/lexical_analysis.html#strings

No Separate Character Type

No separate character type

"characters" are simply one element strings

Python Strings Are Unicode



Source encoding: UTF-8

UTF-8 literals

bytes

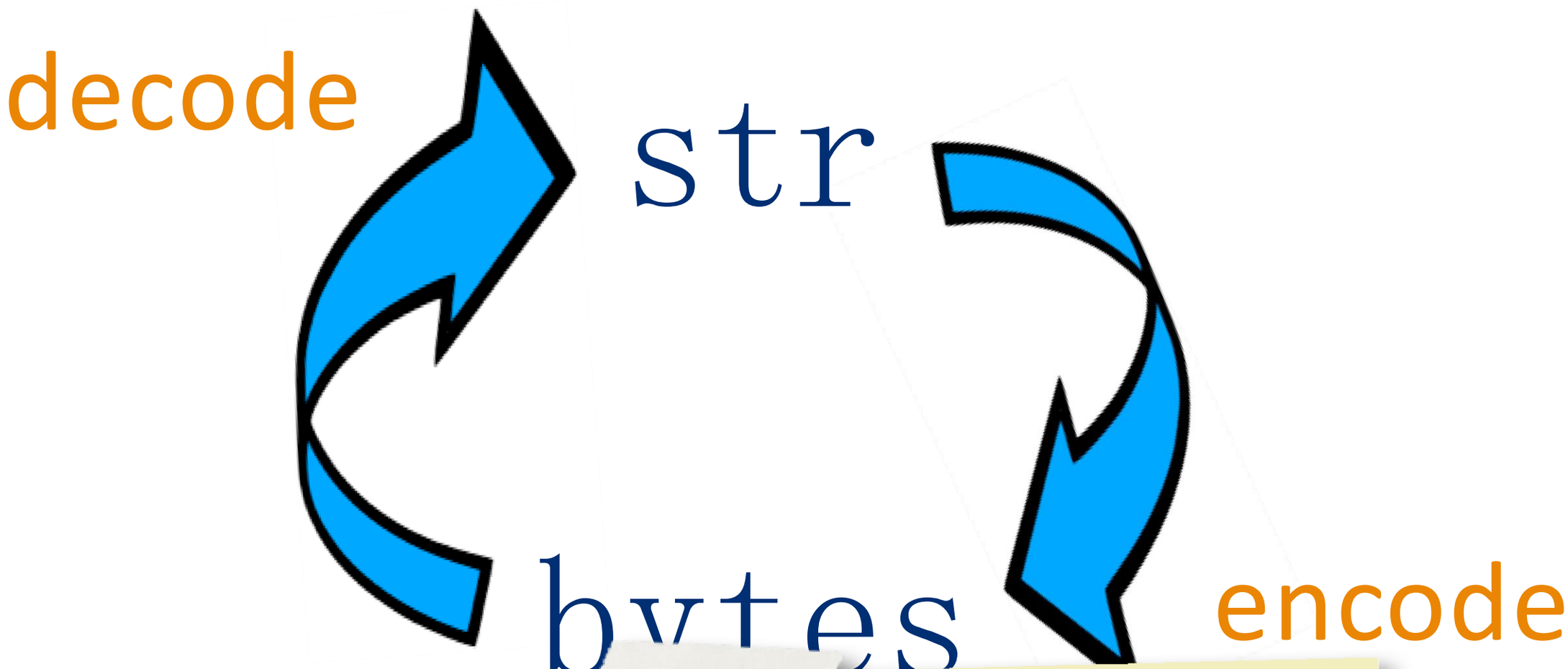
immutable sequences of bytes

Bytes Literals

b ‘data’

b “data”

Converting Between Strings and Bytes



Encodings

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

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()`

python™

Advanced

str

homogeneous immutable sequence of
Unicode codepoints (characters)

str

- `len(s)` gives number of codepoints (characters)

str

- The + operator can be used for string concatenation.
- Strings are immutable, so the += operator re-binds the reference to a new object.
- Use sparingly – concatenation with + or += can cause performance degradation.

str

- Call the `join()` method on the **separator** string
- Use `split()` to divide a string into a list
- Without an argument, `split()` divides on whitespace
- `join()`-ing on an **empty separator** is an important and fast way of concatenating a collection of strings

str

- The `partition()` method divides a string into three around a separator: *pre*fix, *separator*, *su*ffix
- Tuple unpacking is useful to destructure the result
- Use underscore as a dummy name for the separator
- Underscore convention understood by many tools

str

- Use `format()` to insert values into strings
- *Replacement* fi*elds* delimited by `{` and `}`
- Integer field names matched with positional arguments
- Field names can be omitted if used insequence
- Named fields are matched with keyword arguments

str

- Access values through keys or indexes with square brackets in the replacement field.
- Access attributes using dot in the replacement field.
- The replacement field mini-language provides *many* value and alignment formatting options.