Ãfibercool

Outline

- character sets (what)
- encodings (how)
- a bit intertwined
- generalizations
- Squeak specific
- european examples

Character Sets

sets of characters

ASCII

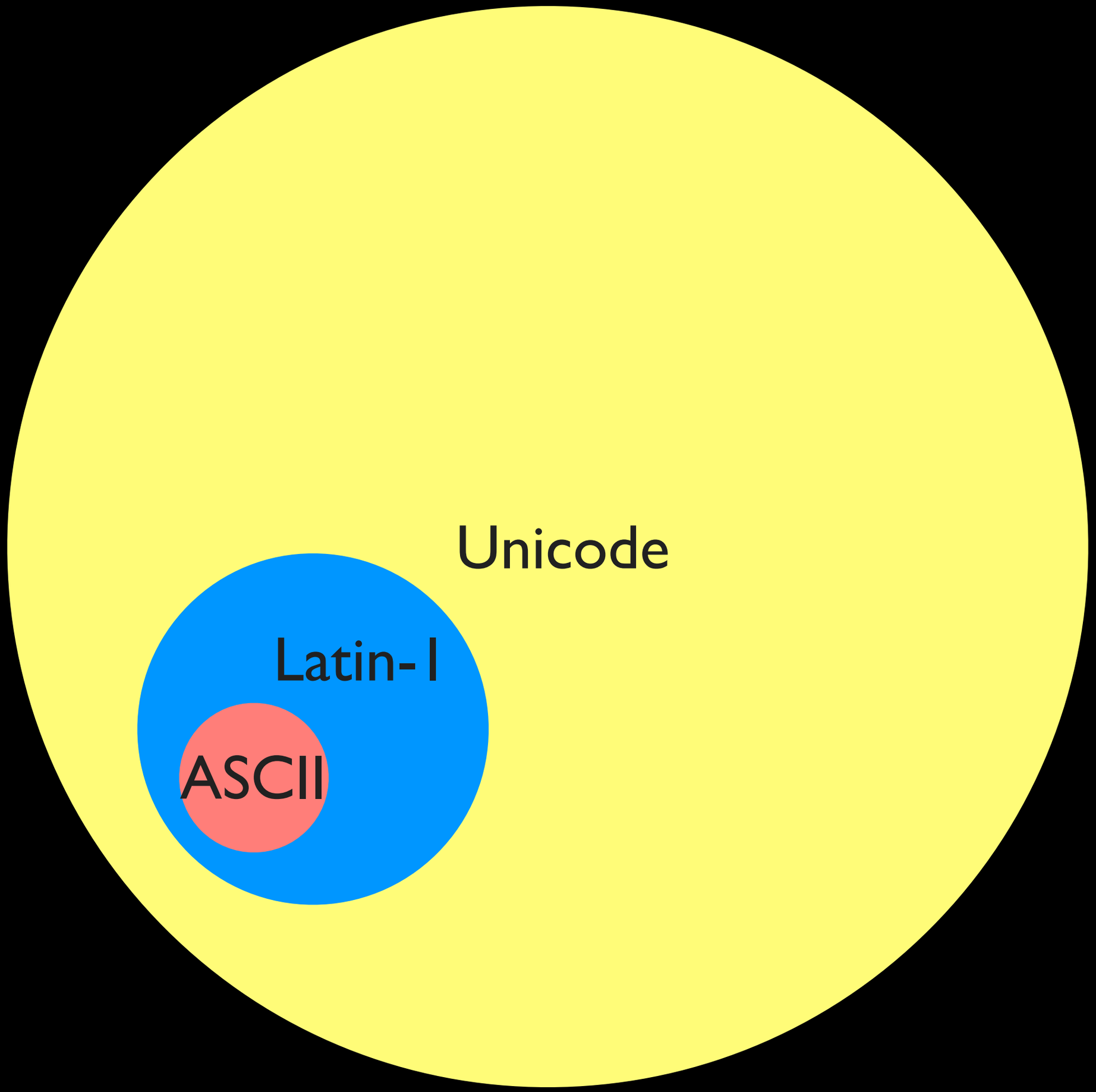
- ~~North America~~, United States
- 128 characters

ISO-8859-1
(latin 1)

- Western Europe
- Does not include € (ISO-8859-15 / latin-9)
- ASCII + 128 additional charters

Unicode

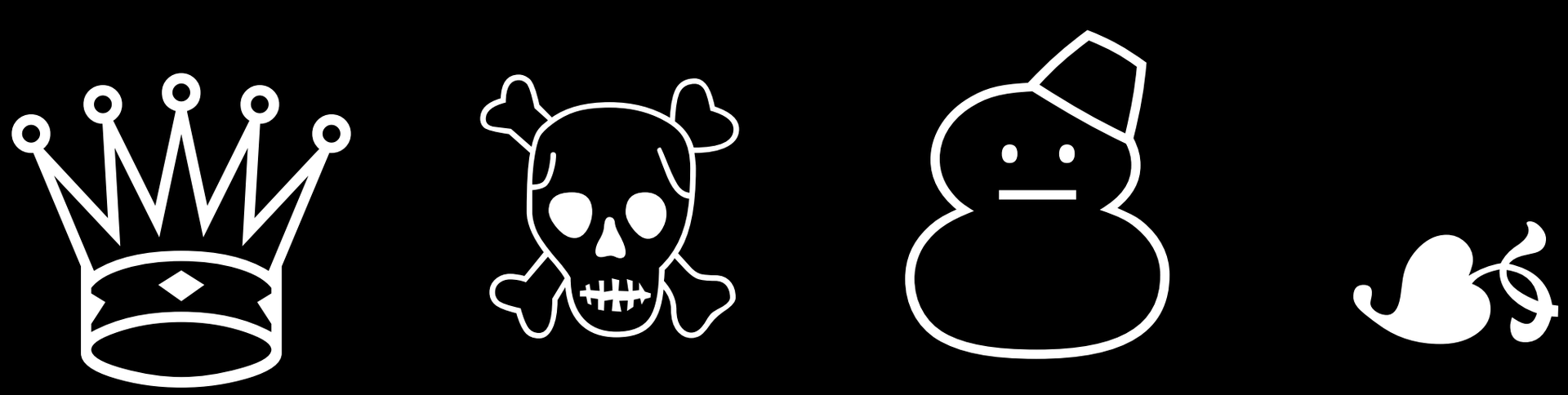
one character set to rule them all


Unicode
Latin-1
ASCII

Code point

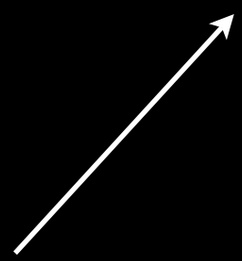
“atom” of text

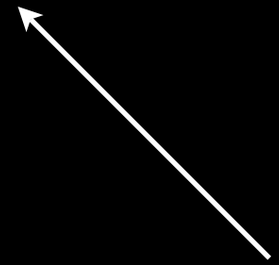
Part of Unicode



NOT Part of Unicode

Integer
(abstract)



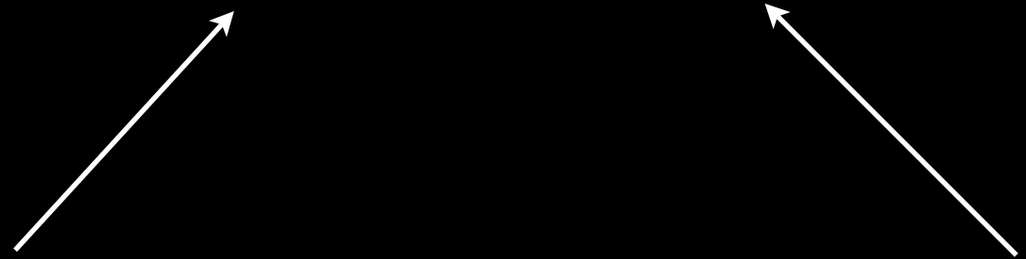


SmallInteger
(-1073741824
1073741823)

LargeInteger
(∞ - SmallInteger)

ranges, no endianness!

String
(abstract)



ByteString
(ISO-8859-1)

WideString
(Unicode - ISO-8859-1)

character set, not encoding!

#leadingChar

- mixes abstraction layers
- language
- presentation

Algorithms

- know Unicode
- know all the rules
- know all the code points
- know all the locales

Transformations

Fußball

FUSSBALL

locale dependent!

Collation (ordering)

- ABC...RSTUVWXYZ
- ÄB...NOÖ...SßTUÜV...YZ
- ABC...RSTUVWXYZÅÄÖ

locale dependent!

Normalization

(what does #= really mean?)

- ¨ + a = ä
- there are different ones to chose from

PHP 6 will do all of this

Encodings

mappings from one space to an other
(isomorphisms)

1:1

- ASCII
- ISO-8859-1

ASCII

- 7 bit
- 8 bit

ISO-8859-1

16rFC

UTF-32

16rFC 16r00 16r00 16r00 (LE)

16r00 16r00 16r00 16rFC (BE)

UTF-16

16rFC 16r00 (LE)

16r00 16rFC (BE)

UTF-8

16rC3 16rBC

WAKom

1:1 direct mapping from bytes to Charaters

WAKomEncoded*

use it with utf-8!

Content-Type: text/html;charset=utf-8

```
<meta content="text/html;charset=utf-8"

    http-equiv="Content-Type"/>
```

2.8: WASession >> #charSet

2.9: /seaside/config

Links

- UTF-8 Sampler
- Favourite Unicode Codepoints
- On the Goodness of Unicode
- Characters vs. Bytes

ubercool