Unicode and UTF-8

Lecture 36

A standard for the discrete representation of written text
The Big Picture

glyphs

characters

code points

binary encoding

code unit
The Big Picture

glyphs

characters

code points

binary encoding
Text: A Sequence of Glyphs

- **Glyph**: “An individual mark on a written medium that contributes to the meaning of what is written.”
  - See foyer floor in main library

- **One character can have many glyphs**
  - Example: Latin E can be e, e, e, e, e, e…

- **One glyph can be different characters**
  - A is both (capital) Latin A and Greek Alpha

- **One unit of text can consist of multiple glyphs**
  - An accented letter (é) is two glyphs
  - The ligature of f+i (fi) is two glyphs
Glyphs vs Characters

glyphs

characters

Latin small E

Latin capital A

Greek capital alpha

A
Security Issue

- Visual homograph: Two different characters that look the same
  - Would you click here: [www.paypal.com](http://www.paypal.com)?
Security Issue

- Visual homograph: Two different characters that look the same
  - Would you click here: www.paypal.com?
  - Oops! The second ‘a’ is actually CYRILLIC SMALL LETTER A
  - This site successfully registered in 2005

- Other examples: combining characters
  - ň = LATIN SMALL LETTER N WITH TILDE
  - ň = LATIN SMALL LETTER N WITH TILDE + COMBINING TILDE

- “Solution”
  - Heuristics that warn users when languages are mixed and homographs are possible
Unicode Code Points

- Each character is assigned a unique *code point*
- A code point is defined by an integer value, and is also given a name
  - one hundred and nine (109, or 0x6d)
  - LATIN SMALL LETTER M
- Convention: Write code points as U+hex
  - Example: U+006D
- As of June 2017, v10 (see unicode.org):
  - Contains 136,000+ code points
    - [emoji-versions.html#2017](https://www.unicode.org/emoji-versions.html#2017)
  - Covers 139 scripts (and counting...)
    - [unicode.org/charts/](https://www.unicode.org/charts/)
Unicode: Mapping to Code Points

glyphs

characters

code points

binary encoding
Organization

- Code points are grouped into categories
  - Basic Latin, Cyrillic, Arabic, Cherokee, Currency, Mathematical Operators, ...

- Standard allows for $17 \times 2^{16}$ code points
  - 0 to 1,114,111 (i.e., > 1 million)
  - U+0000 to U+10FFFF

- Each group of $2^{16}$ called a *plane*
  - U+nnnnnnn, same green $==$ same plane

- Plane 0 called *basic multilingual plane* (BMP)
  - Has (practically) everything you could need
  - Convention: code points in BMP written U+nnnnn (ie with leading 0's if needed)
  - Others written without leading 0's
## Basic Multilingual Plane

<table>
<thead>
<tr>
<th>00</th>
<th>01</th>
<th>02</th>
<th>03</th>
<th>04</th>
<th>05</th>
<th>06</th>
<th>07</th>
<th>08</th>
<th>09</th>
<th>0A</th>
<th>0B</th>
<th>0C</th>
<th>0D</th>
<th>0E</th>
<th>0F</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>1A</td>
<td>1B</td>
<td>1C</td>
<td>1D</td>
<td>1E</td>
<td>1F</td>
</tr>
<tr>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>2A</td>
<td>2B</td>
<td>2C</td>
<td>2D</td>
<td>2E</td>
<td>2F</td>
</tr>
<tr>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>3A</td>
<td>3B</td>
<td>3C</td>
<td>3D</td>
<td>3E</td>
<td>3F</td>
</tr>
<tr>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
<td>4A</td>
<td>4B</td>
<td>4C</td>
<td>4D</td>
<td>4E</td>
<td>4F</td>
</tr>
<tr>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>55</td>
<td>56</td>
<td>57</td>
<td>58</td>
<td>59</td>
<td>5A</td>
<td>5B</td>
<td>5C</td>
<td>5D</td>
<td>5E</td>
<td>5F</td>
</tr>
<tr>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
<td>64</td>
<td>65</td>
<td>66</td>
<td>67</td>
<td>68</td>
<td>69</td>
<td>6A</td>
<td>6B</td>
<td>6C</td>
<td>6D</td>
<td>6E</td>
<td>6F</td>
</tr>
<tr>
<td>70</td>
<td>71</td>
<td>72</td>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
<td>77</td>
<td>78</td>
<td>79</td>
<td>7A</td>
<td>7B</td>
<td>7C</td>
<td>7D</td>
<td>7E</td>
<td>7F</td>
</tr>
<tr>
<td>80</td>
<td>81</td>
<td>82</td>
<td>83</td>
<td>84</td>
<td>85</td>
<td>86</td>
<td>87</td>
<td>88</td>
<td>89</td>
<td>8A</td>
<td>8B</td>
<td>8C</td>
<td>8D</td>
<td>8E</td>
<td>8F</td>
</tr>
<tr>
<td>90</td>
<td>91</td>
<td>92</td>
<td>93</td>
<td>94</td>
<td>95</td>
<td>96</td>
<td>97</td>
<td>98</td>
<td>99</td>
<td>9A</td>
<td>9B</td>
<td>9C</td>
<td>9D</td>
<td>9E</td>
<td>9F</td>
</tr>
<tr>
<td>A0</td>
<td>A1</td>
<td>A2</td>
<td>A3</td>
<td>A4</td>
<td>A5</td>
<td>A6</td>
<td>A7</td>
<td>A8</td>
<td>A9</td>
<td>AA</td>
<td>AB</td>
<td>AC</td>
<td>AD</td>
<td>AE</td>
<td>AF</td>
</tr>
<tr>
<td>B0</td>
<td>B1</td>
<td>B2</td>
<td>B3</td>
<td>B4</td>
<td>B5</td>
<td>B6</td>
<td>B7</td>
<td>B8</td>
<td>B9</td>
<td>BA</td>
<td>BB</td>
<td>BC</td>
<td>BD</td>
<td>BE</td>
<td>BF</td>
</tr>
<tr>
<td>C0</td>
<td>C1</td>
<td>C2</td>
<td>C3</td>
<td>C4</td>
<td>C5</td>
<td>C6</td>
<td>C7</td>
<td>C8</td>
<td>C9</td>
<td>CA</td>
<td>CB</td>
<td>CC</td>
<td>CD</td>
<td>CE</td>
<td>CF</td>
</tr>
<tr>
<td>D0</td>
<td>D1</td>
<td>D2</td>
<td>D3</td>
<td>D4</td>
<td>D5</td>
<td>D6</td>
<td>D7</td>
<td>D8</td>
<td>D9</td>
<td>DA</td>
<td>DB</td>
<td>DC</td>
<td>DD</td>
<td>DE</td>
<td>DF</td>
</tr>
<tr>
<td>E0</td>
<td>E1</td>
<td>E2</td>
<td>E3</td>
<td>E4</td>
<td>E5</td>
<td>E6</td>
<td>E7</td>
<td>E8</td>
<td>E9</td>
<td>EA</td>
<td>EB</td>
<td>EC</td>
<td>ED</td>
<td>EE</td>
<td>EF</td>
</tr>
<tr>
<td>F0</td>
<td>F1</td>
<td>F2</td>
<td>F3</td>
<td>F4</td>
<td>F5</td>
<td>F6</td>
<td>F7</td>
<td>F8</td>
<td>F9</td>
<td>FA</td>
<td>FB</td>
<td>FC</td>
<td>FD</td>
<td>FE</td>
<td>FF</td>
</tr>
</tbody>
</table>

- **Latin scripts and symbols**
- **Linguistic scripts**
- **Other European scripts**
- **Middle Eastern and Southwest Asian scripts**
- **African scripts**
- **South Asian scripts**
- **Southeast Asian scripts**
- **East Asian scripts**
- **Unified CJK Han**
- **Canadian Aboriginal scripts**
- **Symbols**
- **Diacritics**
- **UTF-16 surrogates and private use**
- **Miscellaneous characters**
- **Unallocated code points**
UTF-8

- Encoding of code point (integer) in a sequence of bytes (octets)
  - Standard: all caps, with hyphen (UTF-8)
- Variable length
  - Some code points require 1 octet
  - Others require 2, 3, or 4
- Consequence: Can not infer number of characters from size of file!
- No endian-ness: just a sequence of octets
  - D0 BF D1 80 D0 B8 D0 B2 D0 B5 D1 82...
- Other encodings might not use 8 bits (more general term: code unit)
UTF-8: Code Points & Octets

glyphs
- m
- φ
- ,
- €
- 好

caracters
- Cyrillic ef
- Latin M
- Apostrophe
- Euro sign
- Tei chou ten

code points
- U+006D
- U+0444
- U+2019
- U+20AC
- U+5975

binary encoding
- 6D
- D1
- 84
- E2
- 80
- 99
- AC
- E5
- A5
- BD
UTF-8 Encoding Recipe

- 1-byte encodings
  - First bit is 0
  - Example: 0110 1101 (encodes U+006D)

- 2-byte encodings
  - First byte starts with 110...
  - Second byte starts with 10...
    - Example: 1101 0000 1011 1111
    - Payload: 1101 0000 1011 1111
      = 100 0011 1111
      = 0x043F
    - Code point: U+043F
      *i.e.* п, Cyrillic small letter pe
UTF-8 Encoding Recipe

- Generalization: An encoding of length $k$:
  - First byte starts with $k$ 1’s, then 0
    - Example 1110 0110 ==> first byte of a 3-byte encoding
  - Subsequent $k-1$ bytes each start with 10
  - Remaining bits are payload

- Example: E2 82 AC
  - 11100010 10000010 10101100
  - Payload: 0x20AC (i.e., U+20AC, €)

- Consequence: Stream is self-synchronizing
  - A dropped byte affects only that character
## UTF-8 Encoding Summary

<table>
<thead>
<tr>
<th>Unicode</th>
<th>Byte1</th>
<th>Byte2</th>
<th>Byte3</th>
<th>Byte4</th>
<th>example</th>
</tr>
</thead>
<tbody>
<tr>
<td>U+0000–U+007F</td>
<td>0xxxxxxx</td>
<td></td>
<td></td>
<td></td>
<td>'§' U+0024</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>→ 00100100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>→ 0x24</td>
</tr>
<tr>
<td>U+0080–U+07FF</td>
<td>110yyyxx</td>
<td>10xxxxxx</td>
<td></td>
<td></td>
<td>'¢' U+00A2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>→ 11000010, 10100010</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>→ 0xC2, 0xA2</td>
</tr>
<tr>
<td>U+0800–U+FFFF</td>
<td>1110yyyy</td>
<td>10yyyyxx</td>
<td>10xxxxxx</td>
<td></td>
<td>'€' U+20AC</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>→ 11100010, 10000010, 10101100</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>→ 0xE2, 0x82, 0xAC</td>
</tr>
<tr>
<td>U+10000–U+10FFFF</td>
<td>11110zzz</td>
<td>10zzyyyy</td>
<td>10yyyyxx</td>
<td>10xxxxxx</td>
<td>'颖' U+024B62</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>→ 11110000, 10100100, 10101101, 10100010</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>→ 0xF0, 0xA4, 0xAD, 0xA2</td>
</tr>
</tbody>
</table>

(from wikipedia)
Your Turn

- For the following UTF-8 encoding, what is the corresponding code point(s)?
  - F0 A4 AD A2

- For the following Unicode code point, what is its UTF-8 encoding?
  - U+20AC
Security Issue

- Not all octet sequences are encodings
  - “overlong” encodings are illegal
  - example: C0 AF
    
    \[
    \begin{align*}
    &= \textbf{1100 0000 1010 1111} \\
    &= \text{U+002F (should be encoded 2F)}
    \end{align*}
    \]

- Classic security bug (IIS 2001)
  - Should reject URL requests with “../..”
    - Scanned for 2E 2E 2F 2E 2E (in encoding)
    - Accepted “..%c0%af..” (doesn’t contain x2F)
      - 2E 2E C0 AF 2E 2E
  - After accepting, server then decoded
    - 2E 2E C0 AF 2E 2E decoded into “../..”

- Moral: Work in “code point” space!
Recall: URL encoding

- Concrete invariant (convention)
  - No space, ;, :, & in representation
  - To represent these characters, use %hh instead (hh is ASCII code in hex)
    - %20 for space
  - Q: What about % in abstract value?
- Recall: correspondence relation
Other (Older) Encodings

- In the beginning...
- Character sets were small
  - ASCII: only 128 characters (ie $2^7$)
  - 1 byte/character, leading bit always 0
- Globalization means more characters...
  - But 1 byte/character seems fundamental
- Solutions:
  - Use that leading bit!
    - Text data now looks just like binary data
  - Use more than 1 encoding!
    - Must specify data + encoding used
## ASCII: 128 Codes

### ASCII Code Chart

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>NUL</td>
<td>SOH</td>
<td>STX</td>
<td>ETX</td>
<td>EOT</td>
<td>ENQ</td>
<td>ACK</td>
<td>BEL</td>
<td>BS</td>
<td>HT</td>
<td>LF</td>
<td>VT</td>
<td>FF</td>
<td>CR</td>
<td>SO</td>
<td>SI</td>
</tr>
<tr>
<td>1</td>
<td>DLE</td>
<td>DC1</td>
<td>DC2</td>
<td>DC3</td>
<td>DC4</td>
<td>NAK</td>
<td>SYN</td>
<td>ETB</td>
<td>CAN</td>
<td>EM</td>
<td>SUB</td>
<td>ESC</td>
<td>FS</td>
<td>GS</td>
<td>RS</td>
<td>US</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>!</td>
<td>#</td>
<td>$</td>
<td>%</td>
<td>&amp;</td>
<td>(</td>
<td>)</td>
<td>*</td>
<td>+</td>
<td>,</td>
<td>-</td>
<td>.</td>
<td>/</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>:</td>
<td>;</td>
<td>&lt;=</td>
<td>=&gt;</td>
<td>?</td>
</tr>
<tr>
<td>4</td>
<td>@</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
<td>G</td>
<td>H</td>
<td>I</td>
<td>J</td>
<td>K</td>
<td>L</td>
<td>M</td>
<td>N</td>
<td>O</td>
</tr>
<tr>
<td>5</td>
<td>P</td>
<td>Q</td>
<td>R</td>
<td>S</td>
<td>T</td>
<td>U</td>
<td>V</td>
<td>W</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td>[</td>
<td>\</td>
<td>]</td>
<td>^</td>
<td>_</td>
</tr>
<tr>
<td>6</td>
<td>,</td>
<td>a</td>
<td>b</td>
<td>c</td>
<td>d</td>
<td>e</td>
<td>f</td>
<td>g</td>
<td>h</td>
<td>i</td>
<td>j</td>
<td>k</td>
<td>l</td>
<td>m</td>
<td>n</td>
<td>o</td>
</tr>
<tr>
<td>7</td>
<td>p</td>
<td>q</td>
<td>r</td>
<td>s</td>
<td>t</td>
<td>u</td>
<td>v</td>
<td>w</td>
<td>x</td>
<td>y</td>
<td>z</td>
<td>{</td>
<td></td>
<td></td>
<td>}</td>
<td></td>
</tr>
</tbody>
</table>

4B = Latin capital K
# ISO-8859 family (eg -1 Latin)

<table>
<thead>
<tr>
<th></th>
<th>0001</th>
<th>0002</th>
<th>0003</th>
<th>0004</th>
<th>0005</th>
<th>0006</th>
<th>0007</th>
<th>0008</th>
<th>0009</th>
<th>000A</th>
<th>000B</th>
<th>000C</th>
<th>000D</th>
<th>000E</th>
<th>000F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>.</td>
</tr>
<tr>
<td>1-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>E-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>F-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

| 0-7F match ASCII |

| reserved (control characters) |

| A0-FF differ, eg: |
-1 "Western"
-2 "East European"
-9 "Turkish"
## Windows Family (eg 1252 Latin)

### Windows-1252 (CP1252)

<table>
<thead>
<tr>
<th>x0</th>
<th>x1</th>
<th>x2</th>
<th>x3</th>
<th>x4</th>
<th>x5</th>
<th>x6</th>
<th>x7</th>
<th>x8</th>
<th>x9</th>
<th>xA</th>
<th>xB</th>
<th>xC</th>
<th>xD</th>
<th>xE</th>
<th>xF</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUL</td>
<td>SOH</td>
<td>STX</td>
<td>ETX</td>
<td>EOT</td>
<td>ENQ</td>
<td>ACK</td>
<td>BEL</td>
<td>BS</td>
<td>HT</td>
<td>LF</td>
<td>VT</td>
<td>FF</td>
<td>CR</td>
<td>SO</td>
<td>SI</td>
</tr>
<tr>
<td>DLE</td>
<td>DC1</td>
<td>DC2</td>
<td>DC3</td>
<td>DC4</td>
<td>NAK</td>
<td>SYN</td>
<td>ETB</td>
<td>CAN</td>
<td>EM</td>
<td>SUB</td>
<td>ESC</td>
<td>FS</td>
<td>GS</td>
<td>RS</td>
<td>US</td>
</tr>
<tr>
<td>SP</td>
<td>&quot;</td>
<td>&quot;</td>
<td>#</td>
<td>$</td>
<td>%</td>
<td>&amp;</td>
<td>'</td>
<td>(</td>
<td>)</td>
<td>*</td>
<td>+</td>
<td>-</td>
<td>.</td>
<td>/</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>:</td>
<td>;</td>
<td>&lt;</td>
<td>=</td>
<td>&gt;</td>
<td></td>
</tr>
<tr>
<td>@</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
<td>G</td>
<td>H</td>
<td>I</td>
<td>J</td>
<td>K</td>
<td>L</td>
<td>M</td>
<td>N</td>
<td>O</td>
</tr>
<tr>
<td>P</td>
<td>Q</td>
<td>R</td>
<td>S</td>
<td>T</td>
<td>U</td>
<td>V</td>
<td>W</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td>[</td>
<td>\</td>
<td>]</td>
<td>^</td>
<td>_</td>
</tr>
<tr>
<td>`</td>
<td>a</td>
<td>b</td>
<td>c</td>
<td>d</td>
<td>e</td>
<td>f</td>
<td>g</td>
<td>h</td>
<td>i</td>
<td>j</td>
<td>k</td>
<td>l</td>
<td>m</td>
<td>n</td>
<td>o</td>
</tr>
<tr>
<td>p</td>
<td>q</td>
<td>r</td>
<td>s</td>
<td>t</td>
<td>u</td>
<td>v</td>
<td>w</td>
<td>x</td>
<td>y</td>
<td>z</td>
<td>{</td>
<td></td>
<td></td>
<td>}</td>
<td>~</td>
</tr>
<tr>
<td>€</td>
<td>,</td>
<td>f</td>
<td>&quot;</td>
<td></td>
<td></td>
<td>i</td>
<td>$</td>
<td>Š</td>
<td>Č</td>
<td>Æ</td>
<td>Ž</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>✕</td>
<td>✗</td>
<td>✸</td>
<td>✹</td>
<td>✺</td>
<td>✻</td>
<td>✼</td>
<td>✽</td>
<td>✾</td>
<td>✿</td>
<td>✼</td>
<td>✽</td>
<td>✾</td>
<td>✿</td>
<td>✼</td>
</tr>
<tr>
<td>Ax</td>
<td>NBSP</td>
<td>i</td>
<td>φ</td>
<td>€</td>
<td>£</td>
<td>¥</td>
<td>¢</td>
<td>$</td>
<td>©</td>
<td>a</td>
<td>«</td>
<td>¬</td>
<td>®</td>
<td>‐</td>
<td></td>
</tr>
<tr>
<td>Bx</td>
<td>°</td>
<td>±</td>
<td>³</td>
<td>²</td>
<td>µ</td>
<td>¶</td>
<td>′</td>
<td>ˈ</td>
<td>½</td>
<td>¼</td>
<td>⅓</td>
<td>⅛</td>
<td>‖</td>
<td>‐</td>
<td>‐</td>
</tr>
<tr>
<td>Cx</td>
<td>Á</td>
<td>Â</td>
<td>À</td>
<td>Á</td>
<td>Â</td>
<td>À</td>
<td>Æ</td>
<td>Ç</td>
<td>É</td>
<td>É</td>
<td>É</td>
<td>É</td>
<td>É</td>
<td>É</td>
<td>É</td>
</tr>
<tr>
<td>Dx</td>
<td>Ñ</td>
<td>Ñ</td>
<td>Ñ</td>
<td>Ñ</td>
<td>Ñ</td>
<td>Ñ</td>
<td>Ñ</td>
<td>Ñ</td>
<td>Ñ</td>
<td>Ñ</td>
<td>Ñ</td>
<td>Ñ</td>
<td>Ñ</td>
<td>Ñ</td>
<td>Ñ</td>
</tr>
<tr>
<td>Ex</td>
<td>à</td>
<td>á</td>
<td>å</td>
<td>a</td>
<td>æ</td>
<td>ç</td>
<td>é</td>
<td>è</td>
<td>é</td>
<td>ë</td>
<td>ë</td>
<td>ë</td>
<td>ë</td>
<td>ë</td>
<td>ë</td>
</tr>
<tr>
<td>Fx</td>
<td>ð</td>
<td>ð</td>
<td>ð</td>
<td>ð</td>
<td>ð</td>
<td>ð</td>
<td>ð</td>
<td>ð</td>
<td>ð</td>
<td>ð</td>
<td>ð</td>
<td>ð</td>
<td>ð</td>
<td>ð</td>
<td>ð</td>
</tr>
</tbody>
</table>

**92 = apostrophe**
## HTML 5 Standard

<table>
<thead>
<tr>
<th>Name</th>
<th>Labels</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>The Encoding</strong></td>
<td></td>
</tr>
<tr>
<td>UTF-8</td>
<td>&quot;unicode-1-1-utf-8&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;utf-8&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;utf8&quot;</td>
</tr>
<tr>
<td>windows-1252</td>
<td>&quot;ansi_x3.4-1968&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;ascii&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;cp1252&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;cp819&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;csisolatin1&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;ibm819&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;iso-8859-1&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;iso-ir-100&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;iso8859-1&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;iso88591&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;iso_8859-1&quot;</td>
</tr>
</tbody>
</table>
Early Unicode and UTF-16

- Unicode started as $2^{16}$ code points
  - The BMP of modern Unicode
  - Bottom 256 code points match ISO-8859-1
- Simple 1:1 encoding (UTF-16)
  - Code point $\rightarrow$ 2-byte code unit (16 bits, 1 word)
  - Simple, but leads to bloat of ASCII text
- Later added code points outside of BMP
  - A pair of words (surrogate pairs) carry 20-bit payload split, 10 bits in each word
    - First: $1101\ 10xx\ xxxx\ xxxx\ (x\text{D800-DFFF})$
    - Second: $1101\ 11yy\ yyyy\ yyyy\ (x\text{DC00-DFFF})$
- Consequence: U+D800 to U+DFFF became reserved code points in Unicode
  - And now we are stuck with this legacy, even for UTF-8
Demo

- JavaScript and UTF-16
  Let \( x = "\u{1f916}" \) // robot face
  \( x.length \)
  \( x.charCodeAt(0); x.charCodeAt(1); \)
  \( x.charAt(0); x.codePointAt(0); \)

- Ruby and string encodings
  \( x = "\u{1f916}" \)
  \( x.length \)
  \( x.bytes.map { |b| b.to_s(16) } \)
  \( x.encoding \)
  \( x.encode! \) Encoding::UTF_16
  \( x.bytes.map { |b| b.to_s(16) } \)
Basic Multilingual Plane
UTF-16 and Endianness

- A multi-byte representation must distinguish between big & little endian
- One solution: Specify encoding in name
  - UTF-16BE or UTF-16LE
- Another solution: require byte order mark (BOM) at the start of the file
  - U+FEFF (ZERO WIDTH NO BREAK SPACE)
  - There is no U+FFFE code point
  - So FE FF ➞ BigE, while FF FE ➞ LittleE
  - Not considered part of the text
BOM and UTF-8

Should we add a BOM to the start of UTF-8 files too?
- UTF-8 encoding of U+FEFF is EF BB BF

Advantages:
- Forms magic-number for UTF-8 encoding

Disadvantages:
- Not backwards-compatible to ASCII
- Existing programs may no longer work
- *E.g.*, In Unix, shebang (``#!`, *i.e.* 23 21) at start of file is significant: file is a script
  ```
  #! /bin/bash
  ```
To Ponder

- What is a “text” file? (vs “binary”)
  - Given a file, how can you tell which it is?
- A JavaScript program reads in a 5MB file of English prose into a string. How much memory does the string need?
- How many characters does \( s \) contain?
  
  ```javascript
  let s = . . . . //JavaScript
  assert (s.length() == 7) //true
  ```
- Which is better: UTF-8 or UTF-16?
- What’s so scary about:
  
  ..%c0%af..
Summary

- **Text vs binary**
  - In pre-historic times: most significant bit
  - Now: data is data

- **Unicode code points**
  - Integers U+0000..U+10FFFF
  - BMP: Basic Multilingual Plane

- **UTF-8**
  - A variable-length, self-synchronizing encoding of unicode code points
  - Backwards compatible with ISO 8859-1, and hence with ASCII too