



Announcements

- New deadlines because of snow
 - * Today by 5pm: Labs 4 and 5
 - * Tomorrow by 11pm: Project 1A
 - * A week from tomorrow by 11pm:
Project 1B
 - Thursday, February 7



Announcements

- Quiz 2
 - * Each quiz was different
 - * Your TA will tell you if your



Announcements

- Quiz 3
 - * Thursday and Friday
 - * Chapters 7 and 8 of *Fluency*
 - * Review
 - Questions at end of chapters
 - Answers at back of book
 - Lectures



Announcements

- Chapter 10 for today
- Chapter 18 for Friday



Basic HTML and Nesting

```
<html>  
  <head>  
    <title>Basic HTML</title>  
  </head>  
  <body>  
  </body>  
</html>
```



Basic HTML and Nesting

```
<html>
```

```
  <head>
```

```
    <title>Basic HTML</title>
```

```
  </head>
```

```
  <body>
```

```
  </body>
```

```
</html>
```



Basic HTML and Nesting

```
<html>
```

```
  <head>
```

```
    <title>Basic HTML</title>
```

```
  </head>
```

```
  <body>
```

```
  </body>
```

```
</html>
```



Basic HTML and Nesting

```
<html>
```

```
  <head>
```

```
    <title>Basic HTML</title>
```

```
  </head>
```

```
  <body>
```

```
    <p>Content on the Web page goes  
    here</p>
```

```
  </body>
```

```
</html>
```




Basic HTML and Nesting

```
<html>  
  <head>  
    <title>Basic HTML</title>  
  </head>  
  <body>  
    <p>Content on the Web page goes  
    here</p>  
  </body>  
</html>
```



Basic HTML and Nesting

```
<html>  
  <head>  
    <title>Basic HTML</title>  
  </head>  
  <body>  
    <p>Content on the Web page goes  
    here</p>  
  </body>  
</html>
```



HTML DOCTYPE

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
  Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en"
  xml:lang="en">
<head>
  <meta http-equiv="content-type"
    content="text/html;charset=utf-8" />
  <meta http-equiv="Content-Style-Type" content="text/css"
    />
  <title>An XHTML 1.0 transitional template</title>
</head>
<body>
```



DOCTYPE and Validator

- DOCTYPE helps the Web Browser display your file properly on the Web
- Validator looks for the DOCTYPE
 - * Many versions of HTML
 - From 1.0 to 4.01
 - * Now we're moving to XHTML
 - Compliant with XML
 - We'll look at XML later in course



XHTML 1.0 Rules

- Delete any blank lines or tabs or spaces at top of file.
- Change all tags to lower case.
- Put quotes around all attributes in tags
- Fix deprecated, or obsolete, tags:
 - * Change `` to ``
 - * Change `<i>` to ``
 - * Change `<u>` to ``



XHTML 1.0 Rules

- Fix these three *special* tags so they *self-close*:
 - * Change `
` to `
`
 - * Change `<hr>` to `<hr />`
 - * Change `` to ``



XHTML 1.0 Rules

- Nesting issues:
 - * Tags must be "nested" properly:
 - RIGHT
`<p>Book Title by Author</p>`
 - WRONG
`<p>Book Title by Author</p>`
 - * Right or wrong?
 - `<p>`Nesting means always wrap the text with one set of tags and then wrap that within the next`</p>`



XHTML 1.0 Rules

- Nesting issues:
 - * Don't nest a list within `<p>` tags
 - * Always put `
` within `<p>` or `<div>` or `<a>` tags
 - * An `<a>` tag can be within a `<p>` tag but a `<p>` cannot be inside an `<a>` tag.



Validating XHTML 1.0

- Fix one error, save, upload, refresh, revalidate; many errors will fall away.
- The Validator will read the DOCTYPE and know which version of HTML or XHTML you are using
 - * DOCTYPE helps the Web browser display the page correctly



What's The Plan? Algorithmic Thinking

*Step-by-step directions for
whatever someone, or the
computer, needs to do*



Algorithm

- A precise, systematic method for producing a specified result
- In real life we do this all the time:



Video

- Algorithms



Five Essential Properties of Algorithms

1. Input specified

- * Data to be transformed during the computation to produce the output
- * Must specify type, amount, and form of data

2. Output specified

- * Data resulting from the computation—intended result
- * It is possible to have no output



Five Essential Properties (cont'd)

3. Definiteness

- * Specify the sequence of events
- * Details of each step, including how to handle errors

4. Effectiveness

- * The operations are doable

5. Finiteness

- * Must eventually stop



Language in Algorithms

- *Natural language*
 - * For people, we use a natural language like English
 - * Ambiguity is common in natural language
- *Programming Language*
 - * Formal languages designed to express algorithms
 - * Precisely defined; no ambiguity



Context Matters

- Program can fulfill five properties of an algorithm, be unambiguous, and still not work right because it is executed in the wrong context
 - * e.g., last name in Western countries means family name; in Asian countries it may mean given name
- Context matters: Driving instructions
 - * "From the Limmat River go to Bahnhof Strasse and turn right."
 - * Assumes you are traveling in a specific direction. If you are not, the directions will fail.

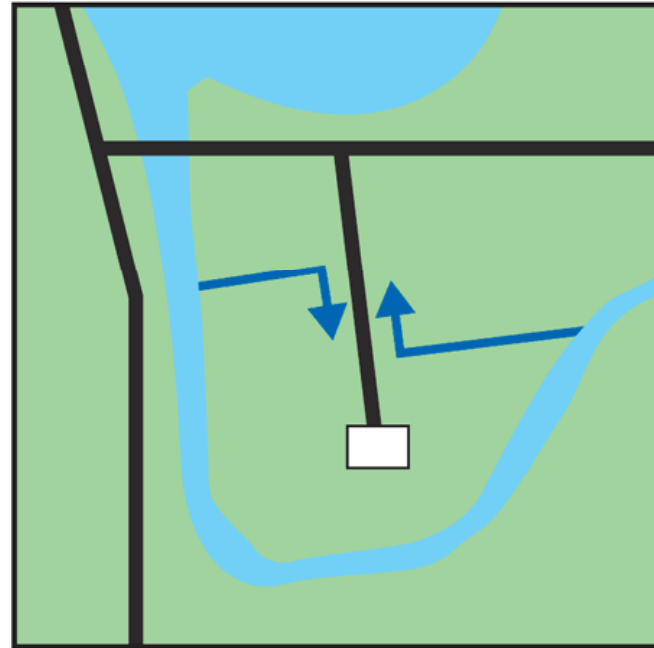


Figure 10.1. Diagram of approaching a street (Bahnhof Strasse) from different directions, giving the “turn right” instruction different meanings.



Program vs. Algorithm

- A program is an algorithm that has been customized to
 - * solve a specific task
 - under a specific set of circumstances
 - using a specific language
- Algorithm is a general method; program is a specific method



An Algorithm: *Alphabetize CDs*

- Imagine CDs in a slotted rack, not organized
- You want to alphabetize by name of group, performing musician, or composer
- How do you solve this problem?



Animation

- Sorting CDs



Analyzing Alphabetize CDs Algorithm

- Illustrates the five basic properties of algorithms
 - * Inputs and Outputs were listed
 - * Each instruction was defined precisely (definiteness)
 - * Operations are effective because they are simple and mechanically doable
 - * Alphabetizing is mechanical, so our algorithm is effective
 - * Finiteness is satisfied because there are only a finite number of slots that can be paired, so instructions 4, 5, and 6 cannot be repeated indefinitely



A Deeper Analysis

- Structural features

- * Two instructions, 5 and 6, in which the agent is directed to go back and repeat instructions. This is called a *loop*.
- * Loops and Tests
 - A loop must include a test to determine whether the instructions should be repeated one more time
- * Assumptions
 - We assume that
 - The CD rack is full (instructions do not handle the case of an empty slot)
 - The word "following" means a slot further from the start point

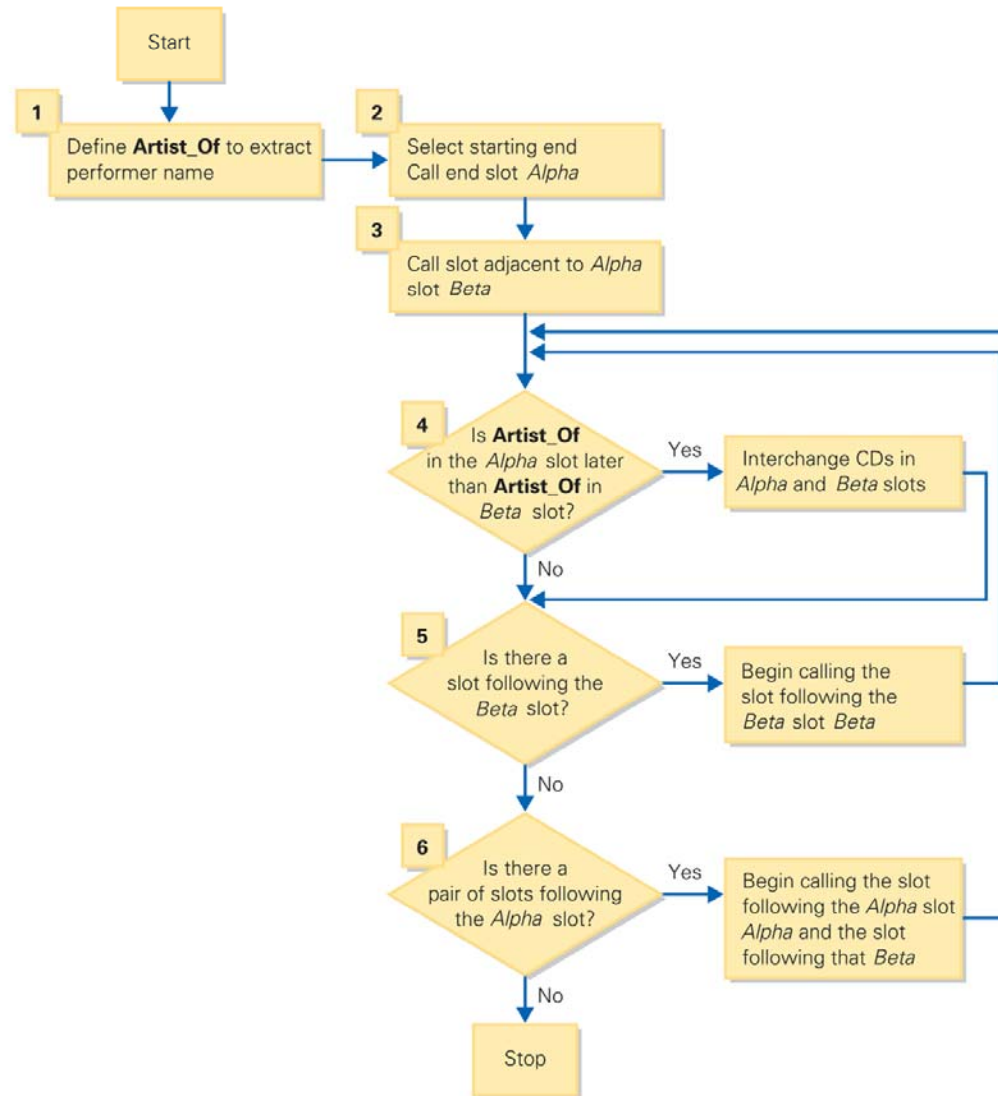


Figure 10.3. Flowchart of Alphabetize CDs. Operations are shown in rectangles; decisions are shown in diamonds. Arrows indicate the sequencing of the operations.



Exchange Sort Algorithm

- The Alphabetize CDs example illustrates the standard *Exchange Sort* algorithm
 - * The idea of comparing pairs of items chosen in a particular way, exchanging them if they are out of order, and continuing to sweep through the items
 - * We could use the same algorithm to sort on a different principle



Announcements

- Chapter 18 for Friday