



EECS 1710

Programming for Digital Media

Lecture 12 :: Arrays [2]

Reverse an array?

```
final int MAX_ELEMENTS = 100;  
int [] myArray = new int[MAX_ELEMENTS ];  
  
// assume array elements are set/assigned to here  
  
// now reverse the order of the elements
```

Reverse an array

```
final int MAX_ELEMENTS = 100;
int [] myArray = new int[MAX_ELEMENTS ];
int [] myArrayReversed = new int[MAX_ELEMENTS ];

// assume array elements are set/assigned to here

// now reverse the order of the elements

int last = myArray.length-1;

for (int i = 0; i<=last; i++)
{
    myArrayReversed[i] = myArray[last-i];
}
```

Sum the values in an array?

```
final int MAX_ELEMENTS = 100;
int [] myArray = new int[MAX_ELEMENTS ];

// assume array elements are set/assigned to here

int sum;

// code to sum elements

println("The total sum = " + sum);
```

Sum the values in an array?

```
final int MAX_ELEMENTS = 100;
int [] myArray = new int[MAX_ELEMENTS ];

// assume array elements are set/assigned to here

int sum = 0;

// code to sum elements

for (int i = 0; i<myArray.length; i++)
{
    sum += myArray[i];
}

println("The total sum = " + sum);
```

Find the maximum element in an array?

```
final int MAX_ELEMENTS = 100;  
int [] myArray = new int[MAX_ELEMENTS];  
  
// assume array elements are set/assigned to here  
  
int indexOfMax;  
int maxValue;  
int currElement; // current element in array
```

```
println("largest value = " + maxValue);  
println("found at i = " + indexOfMax);
```

Find the maximum element in an array?

```
final int MAX_ELEMENTS = 100;
int [] myArray = new int[MAX_ELEMENTS ];

// assume array elements are set/assigned to here

int indexOfMax = 0;           // index of first element
int maxValue = myArray[0];    // first element of array
int currElement;              // current element in array

for (int i = 0; i<myArray.length; i++) {

    currElement = myArray[i];

    if ( currElement > maxValue ) {    // we should update maxValue!
        indexOfMax = i;
        maxValue = currElement;
    }
}

println("largest value = " + maxValue);
println("found at i = " + indexOfMax);
```

Find the minimum element in an array?

```
final int MAX_ELEMENTS = 100;  
int [] myArray = new int[MAX_ELEMENTS ];  
  
// assume array elements are set/assigned to here  
  
int indexOfMin;  
int minValue;  
int currElement;
```

```
println("smallest value = " + minValue);  
println("found at i = " + indexOfMin);
```


Find the minimum element in an array?

```
final int MAX_ELEMENTS = 100;
int [] myArray = new int[MAX_ELEMENTS ];

// assume array elements are set/assigned to here

int indexOfMin = 0;
int minValue = myArray[0];
int currElement;

for (int i = 0; i<myArray.length; i++) {

    currElement = myArray[i];

    if ( currElement < minValue ) {           // we should update minValue!
        indexOfMin = i;
        minValue = currElement;
    }
}

println("smallest value = " + minValue);
println("found at i = " + indexOfMin);
```

Find an element?

- Can we find a number in an `int[]` array?
- Can we count how many times a number occurs?

```
int targetValue = 4;      // target to look for in array
boolean exists = false;   // flag to set if found

for (int i = 0; i<myArray.length; i++) {
```

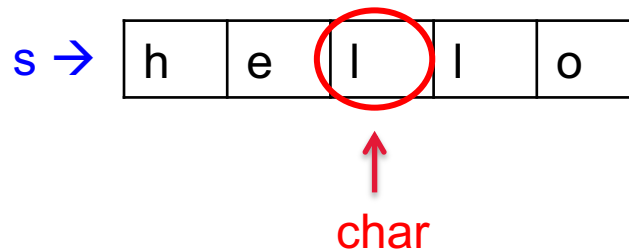
```
}
```

char [] arrays vs. Strings

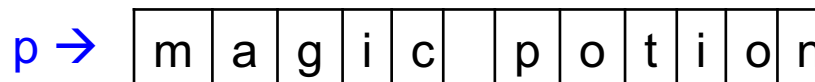
Previously, we had indicated that a String is made up of several characters?

- A String is usually represented as an array of char's !

String s = "hello";



String p = "magic potion";



Strings

- Strings are actually reference types (like arrays)
 - In fact a lot like an array.. of chars
 - + extra attributes & features
- String is actually a special reference type called an Object
 - more next week on Objects
- An Object stores data/properties + methods (together)
 - The methods are specific to the Object

Strings vs. char arrays

- Strings (internally) store an array of chars
- Strings also contain several methods
 - convenient for operating on its array of chars
 - methods are accessed using the “dot syntax”

```
String sentence = "Hello there, how are you?";  
// sentence.methodName() invokes methodName() on this string
```

- there is a method that can actually extract the `char[]` from a `String` object, and work on it directly

```
char[ ]
```

```
toCharArray()
```

Converts this string to a new character array.

Strings vs. char arrays

```
String str = "magic potion";  
char [] strChars = str.toCharArray();
```

```
// find number of 'o's in str??  
int count = 0;  
for (int i=0; i<str.length; i++) {  
  
    if (strChars[i]=='o') {  
        count++;  
    }  
}
```

```
// modify code to find 'o's of any case?  
// modify code to find if an 'o' (any case) exists?
```

Tokenizing a String

```
// IDEA: break up a String into substrings, according to  
// some predetermined delimiter (separating character)  
// use "split" method:
```

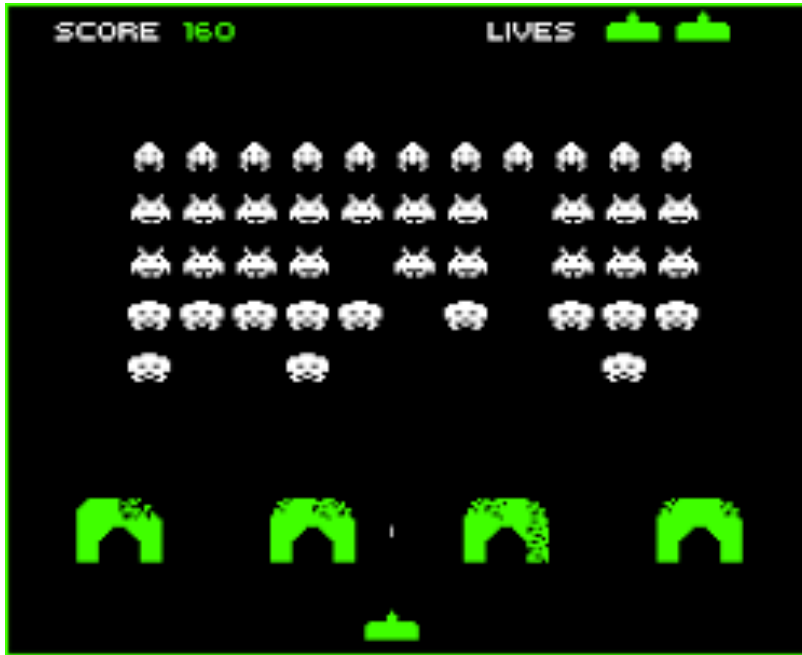
```
String str = "input string 4.5 with words and 32 numbers";  
String[] tokens = split(sentence, ' ');    // delimiter = ' '
```

```
// now tokens is an array of Strings:
```

```
println(tokens[0]);    // prints "input"  
println(tokens[1]);    // prints "string"  
println(tokens[2]);    // prints "4.5"  
println(tokens[3]);    // prints "with"  
...  
println(tokens[7]);    // prints "numbers"
```


Arrays

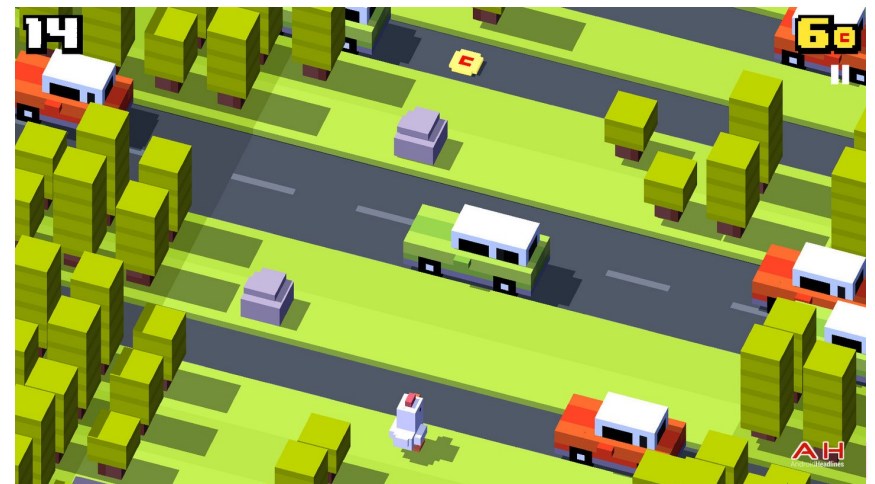
- Very useful for storing/managing **state** (data) in a program
 - Examples:
 - a set of number guesses (number guessing game)
 - the letters in a hangman game
 - a set of shapes/lines drawn to the screen
 - inventory (items collected in a RPG = role playing game)
 - where you are currently on a game board
 - storing moves made or last things drawn (for undo)



Trivia question:

What games are these?

What might relate to *state* here?



Inventory (as an array)

- Common way to store a collection of “things” or “items” that may be used in a game context or similar, is to store them in an array
- Imagine we have a character within a game that can “collect” items (to be used in later parts of the game)
- We will store these in the character’s “inventory”
 - The inventory will be an array of String variables – each entry holding a String “description” of the item

Inventory example

```
final int MAX_ITEMS = 10;
String [] inventory = new String[MAX_ITEMS];
int numItems = 0;

inventory[numItems++] = "banana";
inventory[numItems++] = "stick";
inventory[numItems++] = "BFG";
inventory[numItems++] = "abomb";
inventory[numItems++] = "magic potion"

// output inventory
println("You currently have " + numItems + " items:");
for (int i=0; i<numItems; i++) {
    println(inventory[i]);
}
```

Audio & Images (as Arrays)

- Important area of scientific computing:
Digital Signal Processing (DSP)
- We think of Audio and Images as “signals”
 - Audio: 1D (sound samples over time)
 - Images: 2D (pixel/colour samples over space)

Recap

- Declare and Initialize array (primitives)

```
int[] numbers = { 1, 2, 3 };  
double[] decimals = { 1.1, 3.2, -4.842 };
```

- Declare, then initialize array separately (primitives)

```
int[] numbers;  
numbers = new int[3];  
for (int i=1; i<=3; i++ ) { numbers[i] = i; }
```

- Declare and initialize (non-primitive array)

```
String[] names = { "joe", "jane", "bob" };  
String[] words = new String[3];  
for (int i=0; i<3; i++ ) { words[i] = names[i]; }
```

Audio & Images, as Arrays

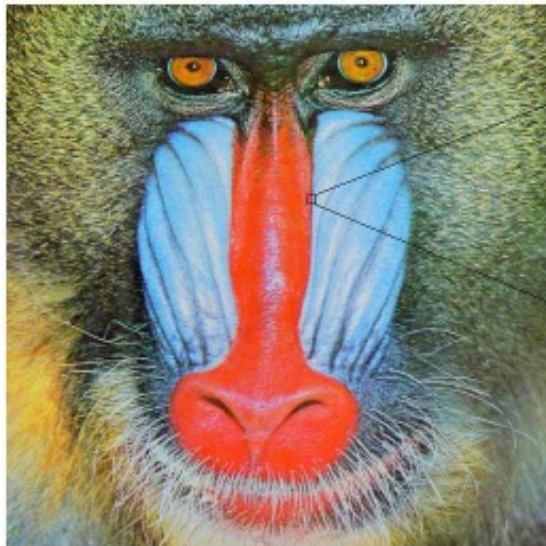
- Audio: 1D (sound samples over time)

1.4	3.5	12	4	0.6	-3.5	-10.3	...
-----	-----	----	---	-----	------	-------	-----

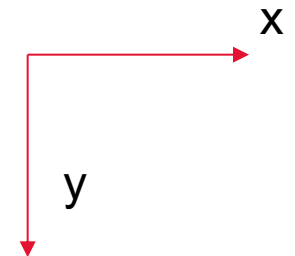
1.4	3.5	12	4	0.6	-3.5	-10.3	...
-----	-----	----	---	-----	------	-------	-----



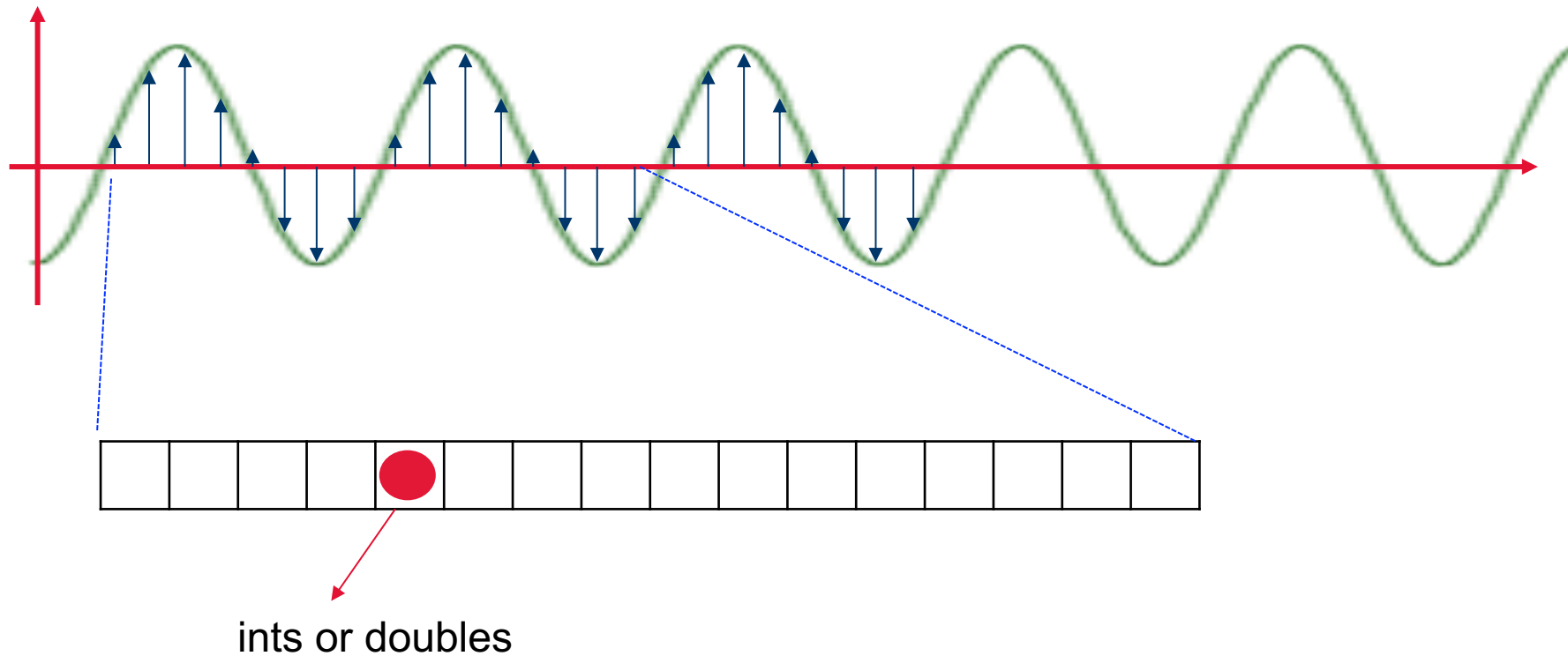
- Images: 2D (pixel/colour samples over space)



(219, 96, 85)	(194, 62, 55)	(147, 174, 219)
(225, 107, 124)	(185, 71, 85)	(135, 166, 216)
(228, 101, 126)	(195, 67, 83)	(144, 185, 226)



Digitized Sound (1D array of voltages)



Digitized Images (2D array of “colours”)

```
int [][] I ;
```

		...				
			I[i][j]			

We will look at 2D arrays later (when we discuss pixel arrays – when working with Images)

