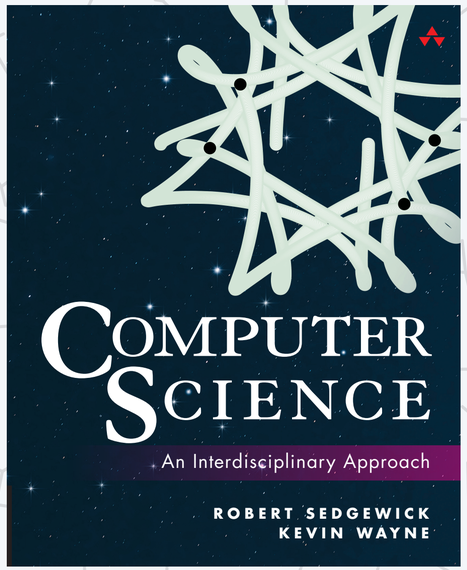
Robert Sedgewick | Kevin Wayne

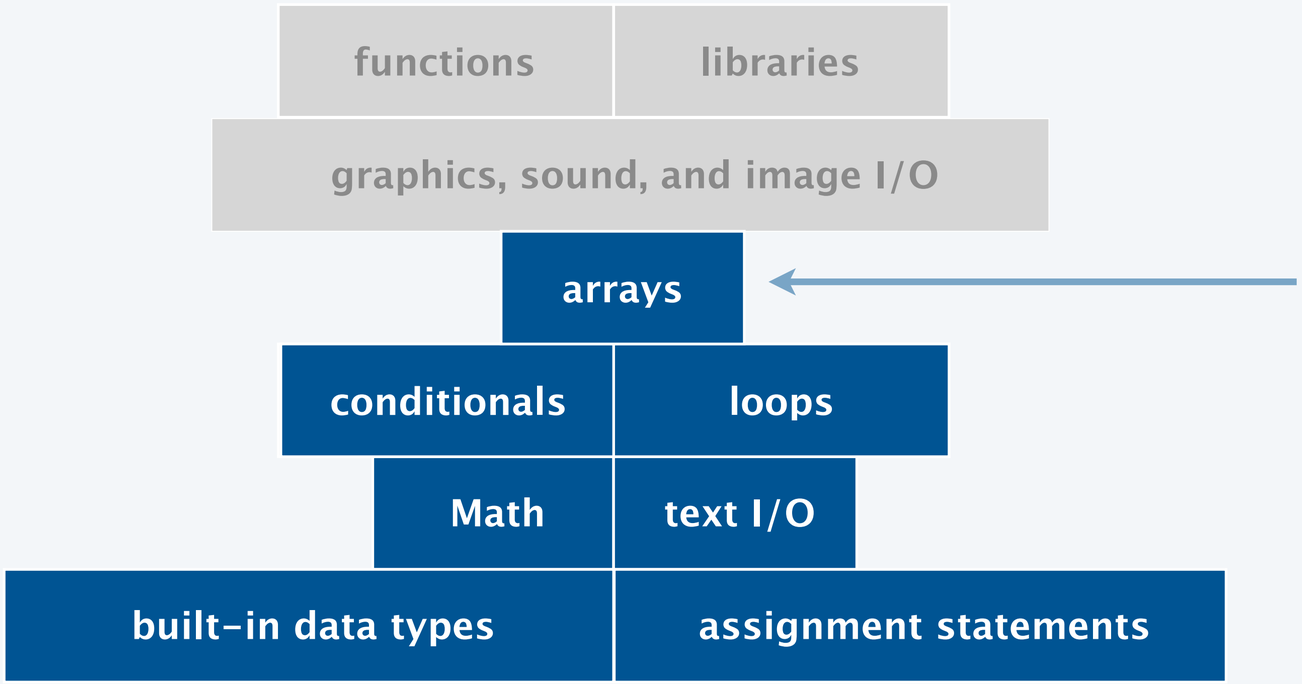


https://introcs.cs.princeton.edu

1.4 Arrays

- *basic concepts*
- *digital audio*

Basic building blocks for programming



we can start storing and processing larger volumes of data

Processing many values of the same type

```
public class AssignmentName {
   public static void main(String[] args) {
      int index = Integer.parseInt(args[0]);

      String name0 = “Hello”;
      String name1 = “Conditionals”;
      String name2 = “Loops”;
      …
      String name5 = “Functions”;

      String name;
      if (index == 0) name = name0;
      else if (index == 1) name = name1;
      else if (index == 2) name = name2;
      …
      else if (index == 5) name = name5;
      else name == “No assignment for this index.”;

      System.out.println(name);
   }
}
```

this part is storing information

this part is processing information

Processing many values of the same type

```
public class PatientName {
   public static void main(String[] args) {
      int id = Integer.parseInt(args[0]);

      String name0 = “Kevin Negy”;
      String name1 = “Marcel Dall’Agnol”;
      …
      String name1259 = “Kevin Wayne”;

      String name;
      if (id == 0) name = name0;
      else if (id == 1) name = name1;
      …
      else if (id == 1259) name = name1259;
      else name = “No patient with this id.”;

      System.out.println(name);
   }
}
```

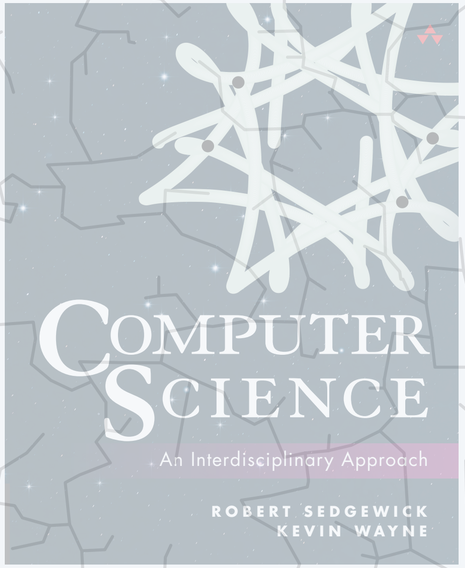
Processing many values of the same type

```
public class PatientName {
   public static void main(String[] args) {
      int id = Integer.parseInt(args[0]);

      String first0 = “Kevin”;
      String last0 = “Negy”;
      …
      String first1259 = “Kevin”;
      String last1259 = “Wayne”;

      String name;
      if (id == 0) name = first0 + “ ” + last0;
      else if (id == 1) name = first1 + “ ” + last1;
      …
      else if (id == 1259) name = first1259 + “ ” + last1259;
      else name = “No patient with this id.”;

      System.out.println(name);
   }
}
```



https://introcs.cs.princeton.edu

1.4 ARRAYS

- *basic concepts*
- *digital audio*

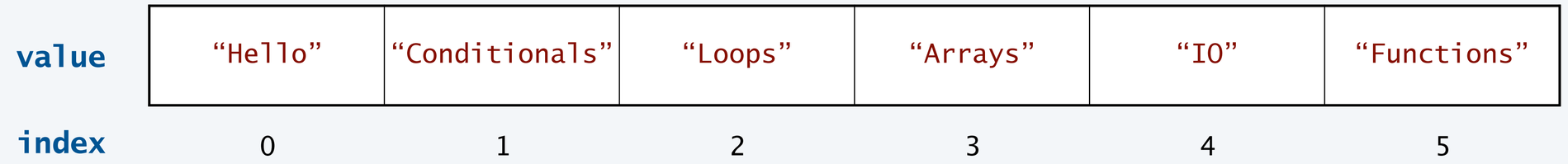
Your first data structure

An array is an *indexed sequence* of values of the same type.

Examples.

- 6 assignments in this course.
- 300 students in COS126.
- 10 million audio samples in a song.
- 4 billion nucleotides in a DNA strand.
- 1 trillion parameters in a large language model.

Main purpose. Facilitate storage and manipulation of data.

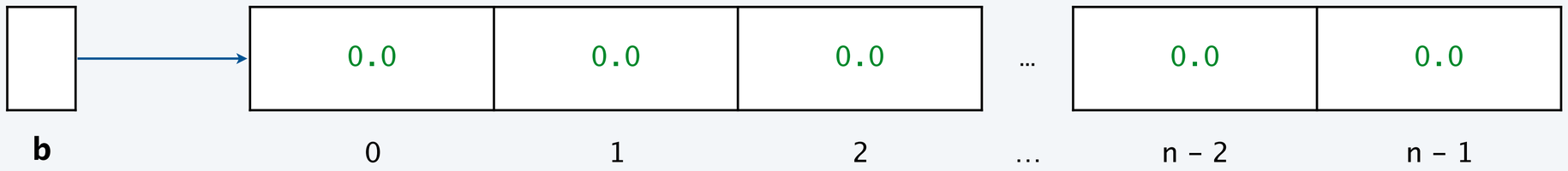


Arrays in Java

Create an array. Specify its type and length.

operation	typical code
declare an array	`double[] a;`
create an array of length ***n***	`a = new double[n];`
declare, create and initialize an array	`double[] b = new double[n];`

all elements are initialized to the default value (zero for numeric values, false for `boolean`*)*

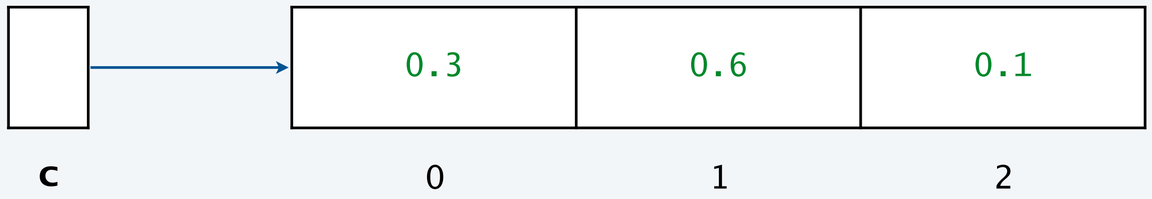


Arrays in Java

Create an array. Specify its type and length.

operation	typical code
declare an array	`double[] a;`
create an array of length n	`a = new double[n];`
declare, create and initialize an array	`double[] b = new double[n];`
array initializer	`double[] c = { 0.3, 0.6, 0.1 };`

all only allowed with declaration:
`c = { 0.3, 0.6, 0.1 }`
causes compile-time error



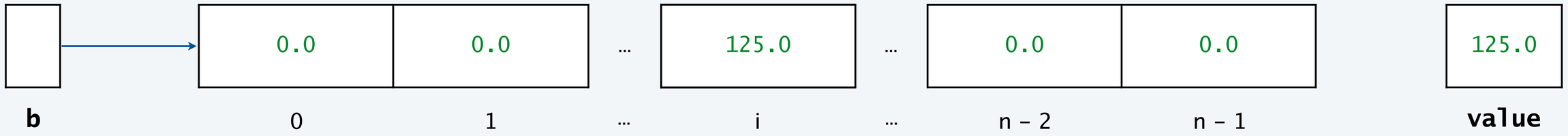
Arrays in Java

Access an array element. Use name of array, square brackets, and index.

operation	typical code
assign 125.0 to `i`-th element	`b[i] = 125.0;`
get array length	`b.length`
assign value of `i`-th element to variable	`double value = b[i];`

index can be any expression of type `int`

array elements are variables (can be used in expressions)



Arrays in Java

problem	code

sum of array elements	`double sum = 0.0;`
	`for (int i = 0; i < n; i++)`
	`  sum += a[i];`

new array with n random numbers	`double[] a = new double[n];`
	`for (int i = 0; i < n; i++)`
	`  a[i] = Math.random();`

months in the year	`String[] MONTHS = {`
	`  "Jan", "Feb", "Mar", "Apr", "May", "Jun",`
	`  "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"`
	`}`

array elements are variables (can be used in expressions)

array elements are variables (can be assigned to)

store predefined constants (used in next assignment!)

Processing many values of the same type

```
double a0 = 0.0;
double a1 = 0.0;
double a2 = 0.0;
double a3 = 0.0;
double a4 = 0.0;
double a5 = 0.0;
double a6 = 0.0;
double a7 = 0.0;
double a8 = 0.0;
double a9 = 0.0;
...
a4 = 3.0;
...
a8 = 8.0;
...
double x = a4 + a8;
```

10 values, without array

tedious and error-prone

```
double[] a = new double[10];

...
a[4] = 3.0;
...
a[8] = 8.0;
...
double x = a[4] + a[8];
```

10 values, with array

easy alternative

```
double[] a = new double[1000000];

...
a[234567] = 3.0;
...
a[876543] = 8.0;
...
double x = a[234567] + a[876543];
```

1 million values, with array

scales to handle huge amounts of data

Arrays: quiz 1

What are the contents of array `a[]` after the loop terminates?

A. A B C D E

B. A B C B A

C. E D C B A

D. E D C D E

E. This code produces an error.

```
String[] a = { "A", "B", "C", "D", "E" };
int n = a.length;

for (int i = 0; i < n / 2; i++) {
    String temp = a[i];
    a[i] = a[n - i - 1];
    a[n - i - 1] = temp;
}
```

Arrays: quiz 2

What are the contents of array `a[]` after the loop terminates?

A. A B C D E

B. A B C B A

C. E D C B A

D. E D C D E

E. This code produces an error.

```
String[] a = { "A", "B", "C", "D", "E" };
int n = a.length;

for (int i = 0; i < n; i++) {
    String temp = a[i];
    a[i] = a[n - i - 1];
    a[n - i - 1] = temp;
}
```

Arrays: quiz 3

What are the contents of array `a[]` after the loop terminates?

A. A B C D E

B. A B C B A

C. E D C B A

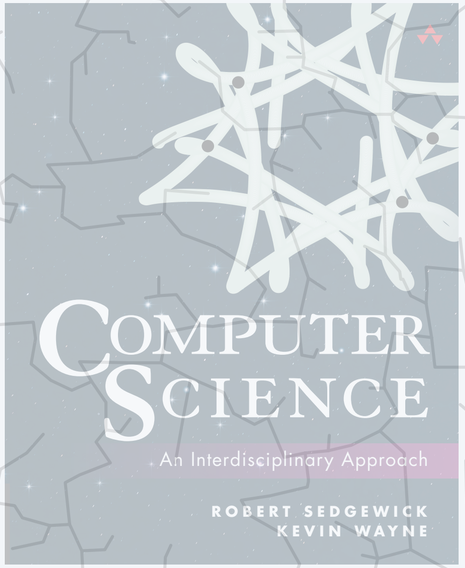
D. E D C D E

E. This code produces an error.

```
String[] a = { "A", "B", "C", "D", "E" };
int n = a.length;

for (int i = 0; i <= n; i++) {
    String temp = a[i];
    a[i] = a[n - i - 1];
    a[n - i - 1] = temp;
}
```


F11
Pause



https://introcs.cs.princeton.edu

1.4 ARRAYS

- *basic concepts*
- *digital audio*

Sound

Sound. The perceptible vibration of air by the ear.

Digital Audio

Audio. An analog or digital encoding of sound.

Audio formats. Vinyl, casette tapes, CD, .wav, .mp3, etc.

Audio signal. Real-valued (between −1 and +1) function of time.

- Value (amplitude) relates to change in sound pressure.

-0.8	0.65	…	0.12	-0.1	

StdAudio library

StdAudio. Our library for playing, reading and saving digital audio. ← *available with `javac-introcs` and `java-introcs` commands*

public class StdAudio	
int SAMPLE_RATE	*44,100 Hz*
void play(String filename)	*play the given .wav file*
void playInBackground(String filename)	*play the given .wav file in a background thread*
void play(double[] samples)	*play the given samples*
void play(double sample)	*play sample*
void save(String filename, double[] a)	*save to a .wav file*
double[] read(String filename)	*read from a .wav file*

Credits

media	source	license
Datacenter	Adobe Stock	education license
Cartoon dinosaur	Adobe Stock	education license
Sound wave	Adobe Stock	education license
Ear	Adobe Stock	education license
Pause	Adobe Stock	education license

Lecture Slides © Copyright 2025 Sebastian Caldas, Marcel Dall'Agnol, Robert Sedgewick and Kevin Wayne