

Alice Handout

- Programming
 - Wikipedia definition http://en.wikipedia.org/wiki/Computer_programming
 - Why Programming Is Fun: <http://www.drdoobs.com/architecture-and-design/why-programming-is-fun/196603871>
 - You are the creator and decide what you want the computer to do. This is empowering and fun. Often you are solving a problem that has no previous solution.
 - Web Programming vs. Computer Programming
- Terminology
 - There are many programming **languages** in the world. Each was created to solve a different problem.
 - Some are evolutions of an existing language (e.g. C++ is a newer version of C).
 - Languages die off when there are no practitioners or tools to support them. Some live on because there are lots of legacy systems that use the language (e.g. FORTRAN and COBOL)
 - **Syntax** is writing "sentences" in the programming language in a way that the compiler/interpreter can read it.
 - Programs operate on **input** from users or other sources (e.g. keyboard, mouse, mobile phone, sensor) and use **output** or display results (e.g. monitor, blinky light)
 - Compilers and interpreters
 - **Compilers** take source code (e.g. a program) and turn it into a computer readable format.
 - **Compiled languages** only run on one type of machine (e.g. C & C++). These programs are much faster because they have been optimized to run fast on that machine.
 - *Example: Think of a language interpreter at the UN. Today his job is to translate an entire speech given by Hillary Clinton into Spanish. If the interpreter was provided the speech text ahead of time, he could translate the entire speech at once and create a hard copy. It would be quick to distribute transcript copies to delegates when they enter the chamber. The work of translation is already done, so when she starts speaking everyone can follow along. Although it takes time up front to translate the whole text, once it is translated it can be given out over and over again (e.g. for delegates who were late to the chamber).*
 - **Interpreters** translate your source code to a machine understandable format on the fly (not ahead of time like compilers).
 - **Interpreted languages** run on any machine (e.g. Ruby & Python). These programs run slower because they aren't optimize for a particular machine.
 - *Example: This time, the UN interpreter has no idea what Hillary will say in her speech. He will need to take each word spoken by Hillary and translate it to Spanish on the fly. This is a much slower process because he will hear the word and translate (input and output at the same time in the same brain). At the end, there is no physical artifact (like a transcript).*
 - **Hybrid** there is a environment/program that is compiled for a particular system (e.g. Java Virtual Machine) but this system interprets an intermediate type of code (e.g. Java byte code). The execution time is somewhere in-between. http://en.wikipedia.org/wiki/Java_virtual_machine
 - **Integrated Development Environment (IDE)** is a bundled set of tools that make programming easier. Examples include Alice, Eclipse, Xcode, and IntelliJ.
 - **Algorithm** is a step by step process. Input is taken, something is done, and output is generated.
- Alice
 - Basics
 - Developed by Carnegie-Mellon to teach programming in a more engaging way.
 - Removes away a lot of the frustration of learning the concepts of coding when you don't have a technical or math background.
 - Instead of typing lines of code (which can be frustrating for new students), you are able to drag objects and code snippets around in the editor to create games and stories.
 - Intro to the IDE and building an opening scene <http://help.alice.org/w/page/55167740/Alice%20Overview>
 - Gallery
 - Objects (characters and props) are arranged in the Scene Editor.
 - The gallery contains dozens of 3D models that can be used to create objects.
 - Each object has its own unique characteristics (properties). The definition of these distinguishing features, the object's properties and the actions it knows how to perform is stored in a file known as a class. As can be seen, each type of object is labeled as a class. Each object in a scene is an instance of its own class.

Alice Handout

- Object Oriented Programming (OOP) http://en.wikipedia.org/wiki/Object-oriented_programming
 - OOP breaks a **system** down into **objects**. This allows the programmer to break a huge complex system up into smaller manageable chunks.
 - **Objects** have **properties** (aka. **state**) and **procedures** (aka. **methods** or **behavior**).
 - **Properties** are used to store information that describe the object (e.g. eye color, height, weight).
 - **Procedures** (aka. **methods** or **behavior**) describes what an object can do (e.g. penguins can waddle).
 - **Inheritance** is used to organize your code.
 - A **parent** has procedures and properties that all of its children will also have (or inherit).
 - **Children** can add their own to what is not present in the parent.
 - **Siblings** are children from the same parent. They both inherit items from the parent, but are different from each other (e.g. Person and Bunny are both Bipedes, but they have very different procedures and properties).
- Programming Foundations
 - Variables
 - Storing a value for use later.
 - There are different types (e.g. text string, whole number, decimal number).
 - **Properties** are special variables that are part of an object. A standard **variable**'s life is fleeting unless it is stored as a property.
 - Arrays
 - A list of similar items (items of the same type).
 - They are stored next to each other in memory, so they all need to be the same size so we know how to find the next one.
 - Methods
 - A collection of statements that performs an action.
 - It's analogous to a recipe. You take input (ingredients) and follow instructions to create output (food!).
 - *Example: You want a bunny to hop. There is no pre-built in function in Alice, so you move the bunny up 2 feet and then back down 2 feet to simulate hopping.*
 - It's a useful tool for organizing your programs and making them more readable. It's especially useful when the action will be performed more than once so you don't need to duplicate code.
 - A method belongs to an object (e.g. AdultPerson.say("Text")).
 - Alice
 - **Procedures** perform an action.
 - **Functions** ask a question.
 - **Properties** fetch or modify a characteristic trait such as name, opacity, and color.
- Control Flow
 - **if** statements are used to make an *either or* decision.
 - **while** loops are loops that keep going until a condition is met. You should make sure the condition will terminate, otherwise you get an **infinite loop** (never stops and eventually crashes your program when you run out of memory).
 - **for** loops are used to loop a specific number of times. Arrays and for loops are often used together because you know how big the array is, and it allows you to loop through each array item.
- Comments
 - I'm a strong advocate!
 - Use them to add information to your code so that you know what you were thinking at the time.
 - They are helpful for organizing your code.
 - Can be used to layout your programming steps before you write the code.