Programming Language (CHP 1)

1. Hypertext Markup Language

It allows the user to create and structure sections, paragraphs, headings, links, and blockquotes for web pages and applications.

HTML is not a programming language, meaning it doesn’t have the ability to create dynamic functionality. Instead, it makes it possible to organize and format documents, similarly to Microsoft Word.

FUN FACTS:

1. The <base>HTML element is quite handy when you have multiple links in your document with the same domain URLs, it allows you to add a base URL to the document which in turn allows you to add only relative URLs to other links in the page as needed.
2. Responsive web development is in vogue with ever growing mobile access. Images can be toggled for different screen sizes with markup. The <picture> HTML5 element lets you add various image sources for different media for the image inside it.

More Facts 🡺 https://www.hongkiat.com/blog/cool-useful-html-tags/

The Creator: HTML was invented by Tim Berners-Lee, a physicist at the CERN research institute in Switzerland. He came up with the idea of an Internet-based hypertext system.

How does HTML Work: HTML documents are files that end with a .html or .htm extension. You can view then using any web browser (such as Google Chrome, Safari, or Mozilla Firefox). The browser reads the HTML file and renders its content so that internet users can view it.

For more information visit https://www.hostinger.com/tutorials/what-is-html

1. Cascading Style Sheets

Remember HTML (Hypertext Markup Language)? The coding language that serves as the foundation for all web development? Well, if HTML is the first language you’ll want to learn when you’re interested in building websites, its cousin CSS is a close second.

The Creator: The W3C invented Cascading Style Sheets (CSS) in 1996 to increase the presentational sophistication and the accessibility of websites, and to eliminate the browser-specific markup that threatened to fragment the emerging web.

How Does CSS Work: CSS brings style to your web pages by interacting with HTML elements. Elements are the individual HTML components of a web page—for instance a paragraph—which in HTML might look like this:

<p>This is my paragraph!</p>

FUN FACTS:

1. The specifications of CSS are maintained by World Wide Web Consortium (W3C).In March 1998, the RFC 2318 registered Internet media type (MIME type) text/css for use with CSS. The W3C conducts a free CSS validation service for CSS documents.
2. Dissimilar to the CSS2 which was comprised of a single document, CSS3 is seperated up into many individual modules which is improving both the functionality and the ease of working.

More Facts 🡺 https://www.geeksforgeeks.org/interesting-facts-about-css/

For more information visit https://skillcrush.com/2012/04/03/css/

1. JavaScript

JavaScript is a very powerful client-side scripting language. JavaScript is used mainly for enhancing the interaction of a user with the webpage. In other words, you can make your webpage more lively and interactive, with the help of JavaScript. JavaScript is also being used widely in game development and Mobile application development.

FUN FACT:

1. JavaScript supports regular expressions in a manner similar to Perl, which provides a concise and powerful syntax for text manipulation that is more sophisticated than the built-in string functions.
2. JavaScript is single threaded. This is the reason lots of people who use multi-threaded programming thinks its working is slow as it would not be able to make use of all the cores of the CPU properly.

More Facts 🡺 https://www.geeksforgeeks.org/interesting-facts-about-javascript/

The creator: JavaScript was developed by Brendan Eich in 1995, which appeared in Netscape, a popular browser of that time. The language was initially called LiveScript and was later renamed JavaScript. There are many programmers who think that JavaScript and Java are the same. In fact, JavaScript and Java are very much unrelated. Java is a very complex programming language whereas JavaScript is only a scripting language. The syntax of JavaScript is mostly influenced by the programming language C.

How Does JavaScript Work: Being a scripting language, JavaScript cannot run on its own. In fact, the browser is responsible for running JavaScript code. When a user requests an HTML page with JavaScript in it, the script is sent to the browser and it is up to the browser to execute it. The main advantage of JavaScript is that all modern web browsers support JavaScript.

For more information visit https://www.guru99.com/introduction-to-javascript.html

History Of The Internet (CHP 2)

The Internet has revolutionized the computer and communications world like nothing before. The invention of the telegraph, telephone, radio, and computer set the stage for this unprecedented integration of capabilities. The Internet is at once a world-wide broadcasting capability, a mechanism for information dissemination, and a medium for collaboration and interaction between individuals and their computers without regard for geographic location. The Internet represents one of the most successful examples of the benefits of sustained investment and commitment to research and development of information infrastructure. Beginning with the early research in packet switching, the government, industry and academia have been partners in evolving and deploying this exciting new technology. Today, terms like “bleiner@computer.org” and “http://www.acm.org” trip lightly off the tongue of the random person on the street.

This is intended to be a brief, necessarily cursory and incomplete history. Much material currently exists about the Internet, covering history, technology, and usage. A trip to almost any bookstore will find shelves of material written about the Internet.

In this paper, several of us involved in the development and evolution of the Internet share our views of its origins and history. This history revolves around four distinct aspects. There is the technological evolution that began with early research on packet switching and the ARPANET (and related technologies), and where current research continues to expand the horizons of the infrastructure along several dimensions, such as scale, performance, and higher-level functionality. There is the operations and management aspect of a global and complex operational infrastructure. There is the social aspect, which resulted in a broad community of Internauts working together to create and evolve the technology. And there is the commercialization aspect, resulting in an extremely effective transition of research results into a broadly deployed and available information infrastructure.

The Internet today is a widespread information infrastructure, the initial prototype of what is often called the National (or Global or Galactic) Information Infrastructure. Its history is complex and involves many aspects – technological, organizational, and community. And its influence reaches not only to the technical fields of computer communications but throughout society as we move toward increasing use of online tools to accomplish electronic commerce, information acquisition, and community operations.

Learn more visit 🡺 https://www.internetsociety.org/internet/history-internet/brief-history-internet/

Coding vs Programming (CHP 3)

“Coding” and “Programming” are the two most important approaches in Software Development Industries. Coding is basically the process of creating codes from one language to another one. It can also be called as a subset of Programming since it actually implements the initial steps of Programming. It involves writing codes in different languages as instructed. Programming is the process of developing an executable machine level program that can be implemented without any error. It is the process of formally writing codes so that the human inputs and corresponding machine outputs remain in sync.

The Difference between Coding and Programming are explained in the below-mentioned points:

1. Coding is the process of translating and writing codes from one language to another whereas Programming is the process of building an executable program that can be used to carry out proper machine level outputs.
2. Coding only deals with the codes and so it is less intimidating and less intensive. On the other hand, Programming deals with a program to control and interact with the machine to produce proper results.
3. Coders are mainly used to translate the requirements and their logic into a language that machines can understand whereas Programming deals with much more than that. It’s not only used to analyze and develop the codes but also engage all the different artifacts to make the system perform in a proper way.
4. Coding is the initial step of developing any software and thus it is much easier and simpler to analyze and understand than Programming. Programming deals with different types of complex scenarios and programs to ensure the proper implementation of the product.
5. Coders only translate the requirement logics into a machine-understandable code without worrying about the details. But on the other hand, Programmers use to analyze and conceptualize different aspects of any program and also solutions to any problems that may or may not occur due to the process. It works on a much broader aspect than coders.
6. To become a programmer, different aspects of any approach needs to be considered. In the case of coding, one just has to deal with the codes and concerned requirements. So, complex programming requires a much more in-depth understanding of the language.
7. Coding can be defined as a part of the Programming approach whereas Programming can be defined as a superset of Coding. It deals with different aspects of any programming base including the coding approach.
8. The machine can’t interact with human communications and it only understands the machine code which is the binary language. So, the main work of a coder is to translate the requirements into machine understandable language.
9. Creating code is the beginning steps and then programming is used to analyze and implement the same and produce the proper machine level output. It also involves all the critical parameters from debugging and compiling to testing and implementation.
10. Coders need to have a thorough understanding of the project working language. However, they mainly code as per the project needs and instructed information. This is the initial step of developing a software product. Programmers use to analyze and conceptualize the different aspects of communication and produce the correct machine outputs. It normally takes much more time for an individual to become a programmer than a coder.

Learn more visit 🡺 https://www.educba.com/coding-vs-programming/

Scientific Method (CHP 4)

A scientific approach involves constant use of the scientific method that has been used by scientists for centuries and is one of the reasons behind the scientific revolution that took mankind from middle ages into space. It rests on the following principles:

1. Coming up with a question
2. Stating a hypothesis
3. Predicting the result
4. Testing it

Analysing the result

So, first you need to formulate a question you are trying to answer. This may sound obvious, but quite often when I sit down to help students having a problem, they can’t say what they are trying to achieve. They may be solving several problems at the same time or solving a problem that they didn’t formulate clearly in their head. Trying to understand what question you’re trying to answer is the important first step.

Then, you need to state the hypothesis that may answer the question. There always are several hypotheses, so use your experience to choose the most likely one. It doesn’t matter if it turns out to be wrong, eventually you will find the right answer. This is the creative part of the process: try several hypotheses to see which one works.

After you have the hypothesis, predict the result you expect. If you skip this step, you won’t know whether your experiment (next step) is successful or not.

Once you know what to expect, do an experiment to test your hypothesis. This is when you write code. This is the only visible step in the process: you actually type something in the computer to see what happens next.

Finally, you’ll get some results back from the computer: it will do what you want, or maybe you’ll see an error message or nothing at all. Even the absence of the result is a result that should be analysed. This is the crucial step that top students don’t skip. When they get an error message they read it carefully and pause to think what could have caused it. Struggling students don’t give an error much thought: they move onto trying something else immediately or just decide that they are “stuck”. If you don’t get the result you expect, you must think why it is so and formulate a better hypothesis (step 2).

This process takes time to describe but it happens almost subconsciously in the head of a developer. If you skip some of the steps, in particular the analysis of the results, you will be just trying random things to see what works. I see this when students just copy-paste blocks of code they don’t understand from the web, hoping that it will work. This is how bad programmers work: they just copy-paste code without giving it much thought.

Learn more visit 🡺 https://blog.makersacademy.com/scientific-method-in-programming-3b729c0b3fc3