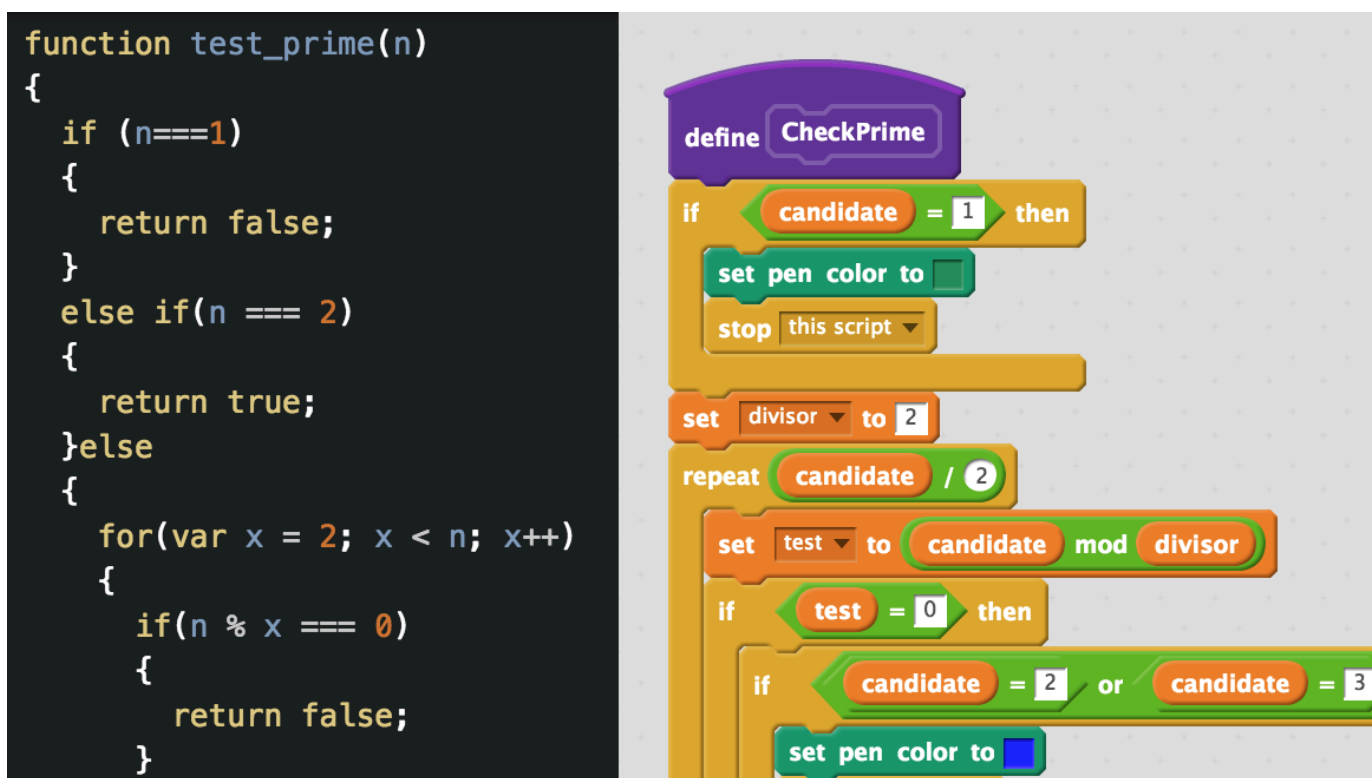


Over the years, knowing how to code—or, at least, knowing the very basics of computer programming—has become an essential skill for applicants looking to join the workforce. Even jobs that are outside the realm of computer science favor applicants with programming knowledge simply because of the skills that programming helps you develop.

And given how widespread computers are in nearly every aspect of society, it's no wonder even children are learning how to code at an early age.

But there's no denying that coding is quite complicated.

What is Block Code?



Block code is generally defined as an element in software programming used to convert text-based software code into a visual block format (see image above and to the right) to minimize potential errors.

This approach allows users to drag and drop visual block representations of text-based code (event listeners, functions, variables, sounds, and more) into the code editor – allowing the user to create games, apps, and other programs.

Block Coding in Coding for Kids

Block coding has made the programming process a lot easier. Because sections of the code are grouped together, it's easier to find issues and address them in text-based coding.

In a similar fashion, block coding utilizes a drag-and-drop approach where groups of code—aka blocks—are already prepared on the side. All the programmer needs to do is select the block he/she wants, drag it to the input field, and drop it.

Once it's been dropped on the field, the coder can then select another group of code (or even the same one) and repeat the process, dragging and dropping all necessary pieces until they've constructed a full sequence that satisfies their goal.

Block coding or block programming is an “entry-level activity” that allows beginners to gain a foundational understanding of how coding works. It also familiarizes them with the sort of computational thinking needed in programming through the use of visual cues rather than pure text.

By giving beginners pre-generated chunks of instructions that they can simply piece together, you eliminate the risk of frustrating syntax and writing-based errors—errors that could potentially frustrate a beginner programmer. Working with blocks makes it easier to just focus on elements such as sequence, situation, problem-solving, and goal construction.

Block Coding: A History

The concept of block coding started in 2003 when MIT developed a basic computer language—the very first block coding language, actually—that was specifically intended to teach kids how to code. The language was known as [Scratch](#).

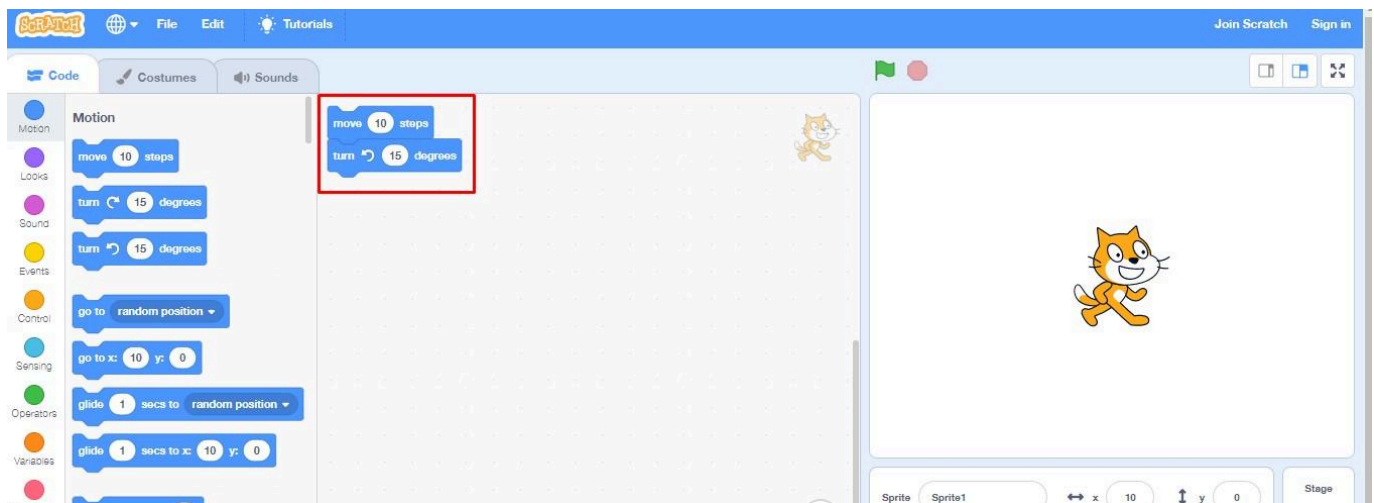
Since then, block programming has been adopted by dozens—if not hundreds—of schools and kids learning programs as the most effective way to teach coding for kids. It relied more on visual aids and practical sequential thinking rather than memorization (which is typically needed when learning text-based coding).

Rather than painstakingly typing out each word, period, or comma, programmers only need to drag, drop, and switch blocks around.

Scratch: Block Coding Samples

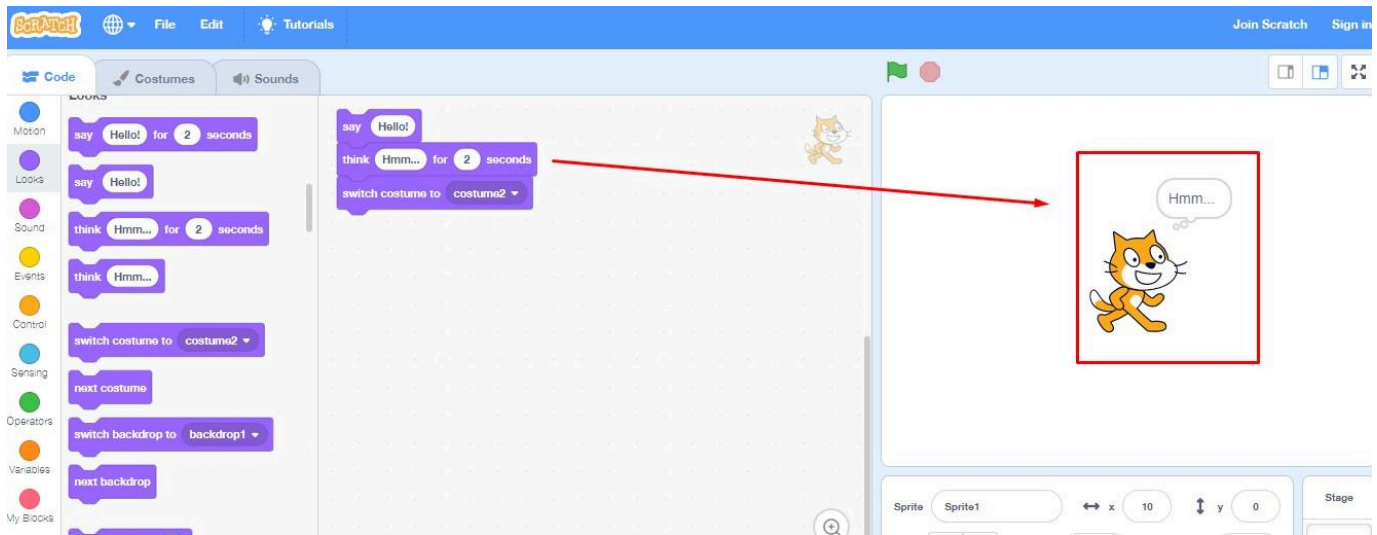
As the first block programming language ever made, Scratch is one of the more popular options when teaching coding for kids via block-based coding. Their library of “blocks” (aka groups or chunks of code and instructions) is pretty extensive, allowing kids to animate colorful cartoon characters (called sprites) however they please. Here are some examples of their block groups:

1. Motion Blocks—controls sprite’s movement; relative to other sprites, the sprite itself, or certain positions (can be set by the programmer)



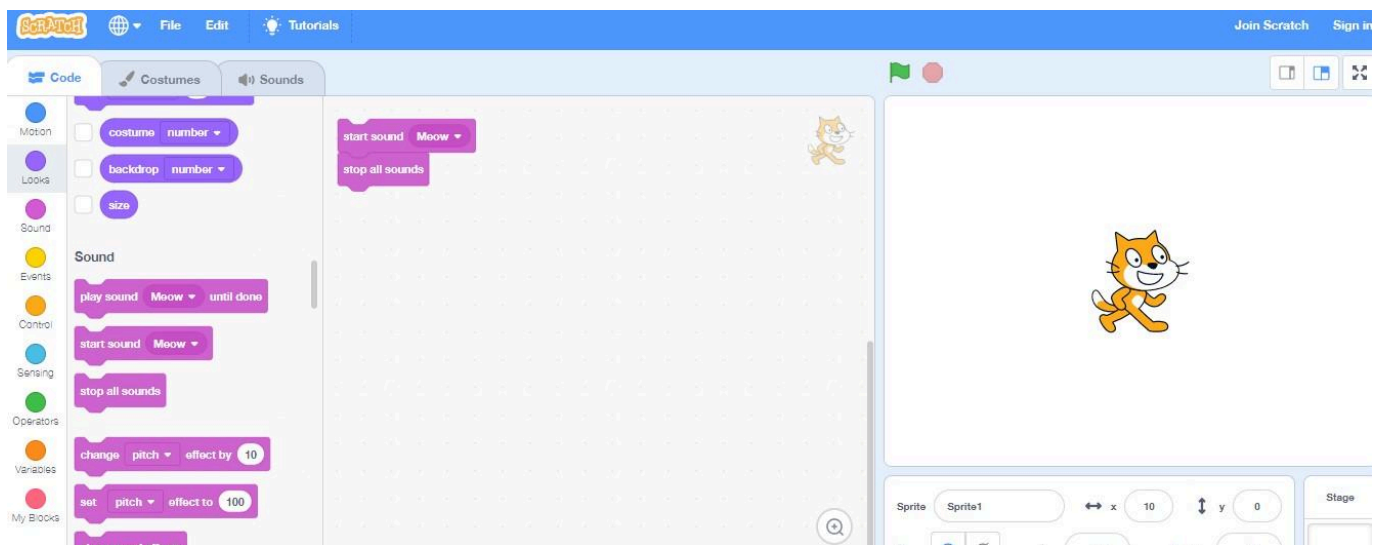
Example: move 10 steps forward, turn 30 degrees to the left

2. Looks Blocks—changes a sprite’s appearance, size, graphic effects; makes a sprite say/think something (using word balloons); changes sprite’s costume or background



Example: say “hello!”, turns the background from blue to yellow

3. Sound Blocks—adds sounds and sound effects; edits pitch or volume



Example: play a “meow” sound, play a “bark!” sound, stop playing the “meow” sound

The Power of Visual, Syntax-Free Programming

Block programming is now considered a branch of “visual programming,” which offsets the typical text-based coding. Block coding is also known for being “syntax-free,” which basically means there’s no need to worry about the specifics of code construction—periods, colons, commas, parentheses, brackets, and the like.

Thanks to these two factors, block coding—or block-based coding—is irrefutably one of the most powerful and effective approaches to teaching programming. And here's why:

Syntax-Free Programming

Drag and drop programming greatly lessens the burden of using complex coding syntax, giving the programmer the headspace he/she needs to focus on finding the best possible solution and implementing it. Using visual blocks in coding eliminates (or, at least, significantly decreases) issues arising from simple syntax errors, consequently cutting down development time.

Converting these blocks to mainstream programming languages for beginners like JavaScript can also be done to create a more extensive and/or detailed script without compromising the foundational code.

Visual Programming

Many people are visual learners, thinkers, creators. A lot of us think in pictures and images rather than words. Appealing to our natural propensity for imagery and visualization often allows us to do our best work. For instance; drawing flow charts on a white board, learning how to troubleshoot a laptop problem by watching a video, or sketching out a quick map to give directions.

Hence why visual programming is so effective. The initial application of computer programming may have started with text-based languages like C++ and Java, but that technical, clinical approach obviously didn't work well for everyone.

Visual programming was created to fulfill the demand of a more human-friendly coding option. It was structured to make coding more accessible at three different levels; (1) syntax, (2) semantics, and (3) simulation.

1. Syntax – eliminate syntax errors by using blocks, icons, forms, and diagrams
2. Semantics – explain the program using meta-information
3. Simulation – check how the program behaves in a certain state by including a visual mechanism to simulate the triggers

Block Coding: Advantages and Disadvantages



Although this drag and drop has some drawbacks, it's hard to deny just how far the benefits outweigh them. However, for the sake of discussion, we'll be listing both here.

Block Coding Advantages

Low Barrier of Entry

This drag-and-drop approach to coding makes it so that literally anyone can do it. There's no need to memorize anything (like functions or sentence structure) or type anything down. Setup is minimal and requires little to no preparation.

It's simple enough for you to understand it with minimal instruction and teachers to introduce it with minimal study and prior knowledge.

Focuses on Programming Concepts

Instead of highlighting proper syntax and sequencing, block coding allows beginners to focus on the fundamental concepts of programming: operators, control structures, commands, loops, conditions, and so on.

Encourages Experimentation

Good programmers can create new code and unique commands by experimenting with existing concepts. However, in the case of text-based coding, the risk of syntactical errors ruining the foundational code can discourage coders from playing around—especially if the coders are just beginners.

Block programming allows kids and coding beginners to experiment with the existing script without the risk of starting over from scratch. And without the option to edit individual lines of code, the chances of dismantling the entire script are pretty low.

Minimal Learning Frustration

Syntax errors can be incredibly frustrating when learning how to code or teaching coding. It can get in the way of conceptualizing a solution in programming, forcing students to pay attention to the text details rather than the coding concept or bigger problem at hand. What's more, constant frustrations during learning can lead to reproach or resentment for the subject matter.

Block Coding Disadvantages

Limited Debugging Options

Lack of syntax and an option to edit individual lines means that you can't really get into the nitty-gritty of the script to see what's wrong. While block coding normally yields very few errors, it doesn't eliminate the errors completely. This means that should issues with the code arise, it's impossible to access the individual commands to source (and potentially fix) the problem.

A step-by-step debugging is simply not possible when most of the steps are grouped together in blocks that cannot be broken down.

Block-Based to Text-Based Conversions

Perhaps the biggest issue most people see with block coding is that “cushions” or “protects” first-time programmers from experiencing the common problems and frustrations that come with coding.

Syntax-free programming eliminates the need to instantly address syntactical errors when they arise.

Visual blocks and cues might keep you from developing or learning mental visualization.

Pre-generated blocks of code eliminate the need to learn and memorize actual syntactical commands.

These same blocks of code also eliminate the chance to use lines of code or commands outside of the pre-generated script.

The Verdict

Block programming is an excellent stepping stone to learning text-based coding. It provides foundational knowledge and encourages an understanding of programming fundamentals. Once you have completely mastered block coding, it's much easier to learn text-based programming as you advance.