

Bookmark a few tricks to help you master a programming language

Learning programming can be extremely stressful if you don't approach it with the right thoughts and attitudes.

Programming is difficult , there is no doubt about it. If someone ever whispers you ' *Oh, programming is very simple* ' then don't believe it! They are people who have many years of experience in the industry so they can make a difficult task like code easy easy. It is normal for you to feel overwhelmed when you first come into contact with programming, because there are so many things to learn, and you may forget that knowledge very quickly after learning them. This is still normal.

However, normal does not mean that it is not frustrating and frustrating. Truth be told, programming can be extremely stressful if you don't approach it with the right thoughts and attitudes.



So how to remember a new language, library or framework quickly and permanently? Please share some of the tips that TipsMake.com has bookmark can help you remember better knowledge, 'death' information acquisition situation enters this ear and comes out in the other ear. Let's follow it!

'Death' method of cramming knowledge

Sure, **knowledge cramming is the worst way to learn programming** . Remember, cramming is probably your way of learning the whole time students and college students are. That's the only way you know how to learn.



However, many studies have shown that the **more you try to cram, the less you remember** . That's the right thing for any academic subject and is especially accurate when it comes to programming.

This statement stems from a 2008 study by the University of California, San Diego:

Students will get better results when they arrange their sessions into sessions than when they try to cram everything into their brains at the same time.

Why? Most likely due to serial position effect:

Serial position effect is a tendency for a person to remember only the first and last part of a sequence while the middle part does not remember well.

In other words: in any class, you are often able to remember the information learned at the beginning and end of the class and forget the information in the middle of the session.

So the goal is that we need to maximize the amount of knowledge of the start and end stages. That is why it is better to break the lesson into short sessions, which will help you remember and absorb new knowledge better.

The next time you sit down to learn the code, just study one topic at a time and have a few minutes of short breaks between each topic. However, be careful not to let that rest period make you fall into a lazy state to delay your studies.

Practice regularly: Mix theory and practice

Another reason to prove that cramming cannot help you remember long-term knowledge is that memory is often faded over time.

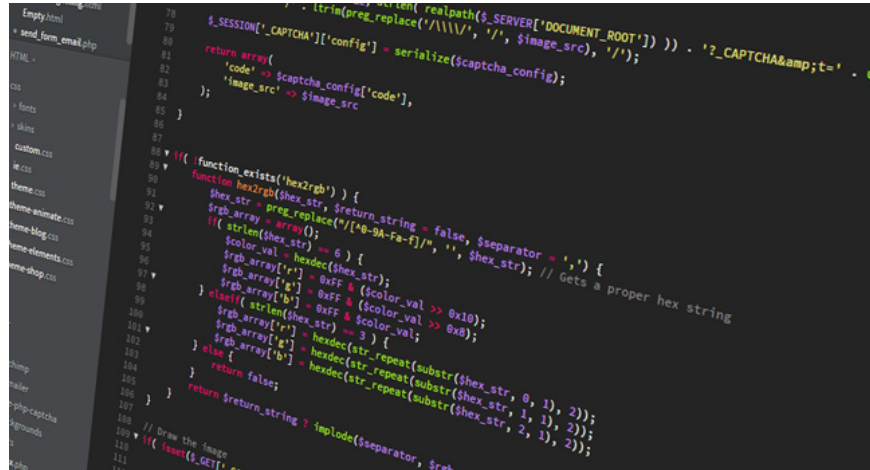
The most important thing to remember when learning programming: it will not be enough if you study a topic only once or twice. You have to review it dozens or even hundreds of times. Each of these reviews will help you inculcate the subject a little more into the brain.

This can be difficult if you're a crammer. However, change your habit of learning badly, you will be surprised because the amount of knowledge in your head will be kept longer through regular review and practice efforts.

Use many different resources

The most difficult aspect of programming, especially for new developers, is that details and nuances need to be acquired. Until knowledge is completely received, you will fall into a state of scratching your hair very much.

Depending on the language, you will need to remember hundreds of syntax rules including keywords, semicolons, whitespace. Some languages have very strict requirements, others are simpler, and others there are languages that have their own unique approach that the syntax you have previously learned cannot be used. All of this can be confusing, confusing if you have not had previous programming experience.



```
78 // ...
79 $SESSION['CAPTCHA'] = array('code' => $captcha_config['code'],
80                             'image_src' => $image_src);
81
82 return array(
83     'code' => $captcha_config['code'],
84     'image_src' => $image_src
85 );
86
87
88 // ...
89 function_exists('hex2rgb') ? {
90     $hex_str = preg_replace('/[0-9A-Fa-f]/', '', $hex_str); // Gets a proper hex string
91     $rgb_array = array();
92     if (strlen($hex_str) == 6) {
93         $color_val = hexdec($hex_str);
94         $rgb_array['r'] = 0xFF & ($color_val >> 0x10);
95         $rgb_array['g'] = 0xFF & ($color_val >> 0x0);
96         $rgb_array['b'] = 0xFF & ($color_val >> 0x0);
97     } elseif (strlen($hex_str) == 3) {
98         $rgb_array['r'] = hexdec(str_repeat(substr($hex_str, 0, 1), 2));
99         $rgb_array['g'] = hexdec(str_repeat(substr($hex_str, 1, 1), 2));
100        $rgb_array['b'] = hexdec(str_repeat(substr($hex_str, 2, 1), 2));
101    } else {
102        return false;
103    }
104    return $return_string ? implode($separator, $rgb_array) : $return_string;
105 }
106
107 // Draw the image
108 // ...
109 // ...
110 // ...
111 // ...
```

Therefore, an explanation is often not enough to capture a particular topic. It is important that you consult a lot of documents, fully exploit the many resources you can. You can refer to categories such as documents, guides, videos, lectures, blogs. Each resource will give you more insight into many aspects of the problem.

In addition, **no programming material is standard**. Everyone praised document A as the best way to learn X language, but it may not make sense to you. Maybe people hate B documents, but when you refer to it, it's amazing.

That's why you need to be ready to learn from all kinds of resources available. Each person learns in a different way. If you're having trouble with a topic, look for more resources.

It is useful to find a mentor or a friend to learn programming. Self-study is great, but it would be better to have someone you can ask questions and get answers quickly. This can turn a frustrating few days into a simple conversation in about 5 minutes, then the problem.

Teach concepts after you learn them

There is a pretty interesting concept in programming called **Debug with rubber duck - Rubber Duck Debugging**. Here is a method to debug the code but not the type done on the IDE but you will take a rubber duck and explain the code to it, explain each line, talk to it when encountering cryptic code as well as venting anger when the code has errors.



Benefits of Rubber Duck Debugging are to help you understand the problem better. When processing a code or reading a topic, you may think you already understand the problem, but it is not sure. Only when you can explain the code yourself clearly in the way you understand it will be considered successful.

You can implement code interpretation method with a certain friend, but the problem is that not everyone is always free to listen to your explanation so using a symbolic object like a duck would be more reasonable. At first it may feel a bit silly, but you will realize that it is extremely effective for remembering knowledge.

This method is called learning by teaching and is basically a variation of the aforementioned Rubber Duck Debugging technique.

In addition, there are many ways to do this. You can write a blog post explaining newly learned concepts, recording videos that explain something you have learned. Each of these will reinforce your knowledge as well as expose the gap in understanding to help you quickly gain knowledge.

Practice properly making great skills

How you practice is far more important than the amount of time you spend practicing.

Reading is passive. Watching tutorial videos on YouTube is also passive. Listening to podcasts is also passive. As a newbie, you can be tempted and jump from one guide to another, soaking up one topic after another but not really applying any of that knowledge in a practical way. Resist this temptation.

If you want to speed up your learning process, then you have **to be ready to learn in an initiative rather than passive** . Positive practice will determine the final result.



Challenge yourself with personal projects. Think about some of the project ideas you want to implement. If you are a newbie who is learning programming, you can create a Tic-Tac-Toe or Hangman game. For seasoned developers try to understand a new framework, or try a simple mobile app or web game. As long as it creates excitement, do it!

With this method you will gain two benefits.

1. First, it will **maintain your focus**. Studies have shown that students will learn better when pursuing topics that interest them. That is exactly what an individual project brings. You have a final goal that you really want to achieve, so you have the ability to retain that information and knowledge.
2. Second, **you have no pressure to succeed**. Success is great, but it doesn't need to be taken seriously because it allows you to be tested and creative. Surely you will encounter problems but will bring a lot of fun and no stress.

Relax & bookmark everything

No one can remember everything they learned. Even if you work with a specific library or framework for a while, it is not dangerous to forget some functions or syntax naturally.

In fact, trying to remember everything will only take your time and effort. The references that exist are also by this reason. Why cram all the encyclopedia knowledge into memory when you can flip it open whenever you need it?



So when should I remember and when should I consult?

For concepts, you should remember them. For all other things such as specific names of functions, parameter lists . it is not necessary to remember details. Keep them in reference documents so that when you don't remember you can open them again easily.

It must be confirmed again: programming is difficult and normal if you have to wrestle with it. Don't torment yourself if you can't remember the knowledge when you first started. Moving forward and constantly learning. Hopefully the advice above is somewhat helpful to you.

You finished reading the article "**Bookmark a few tricks to help you master a programming language**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.