

جديد!

12 قواعد ل تعلم البرمجة

بعد تدريس مليون طالب ، إليك أفضل الطرق لتعلم البرمجة

دكتور. أنجيلا يو

مطور خبير &

مدرس رئيسي في App Brewery

تم نشر



The
App Brewery

طاولة محتويات

خداع دماغك باستخدام قاعدة الـ 20 دقيقة

رمز الغرض

لا توجد لغة "مثالية" للتعلم

افهم ما تكتبه

لا بأس أن لا أعرف

كن مقلدا

كن مسؤولاً

استمر في التعلم

لعب فووسبال

احصل على مرشد

انغمس في عادة التقسيم

كسر رمز شخص آخر



قاعدة واحدة

خدع دماغك في 20 دقيقة

قاعدة



The
App Brewery

تعلم البرمجة يشبه إلى حد ما الذهاب إلى صالة الألعاب الرياضية.

حتى لو تجاوزت الحد الأقصى وقضيت عطلة نهاية أسبوع كاملة في صالة الألعاب الرياضية ، فلن ترى فرقاً واضحاً في جسمك. كلما تعلمت البرمجة بانتظام ، زادت احتمالية أن تبدأ في رؤية عضلات التشفير الممزقة. (لا تضع والمفارقة على لي).



لكن المشكلة أين تجد الوقت؟ بين العمل في وظيفتك بدوام كامل ، وقضاء الوقت مع عائلتك ومسؤول حياتك ، متى من المفترض أن تجلس وتتدرب على هذا "الترميز اليومي"؟

بينما كنت أعمل كطبيب ، قضيت حوالي 12 ساعة في المستشفى ، وساعة واحدة في التنقل وحوالي ساعتين في الأشياء العامة التي تحافظ على الحياة ، مثل تناول الطعام. لذلك لم يتبق لي سوى 9 ساعات في يومي. نظرياً ، يمكن تخصيص ساعتين لممارسة الترميز و 7 ساعات للنوم. ولكن لا يوجد شيء أصعب من محاولة إقناع دماغك المشبع بالعمل بالجلوس والتعرف على الوقت الذي يمكنك فيه مشاهدة لعبة Game of Thrones مع وعاء من الأيس كريم.

واكنر بعد ذلك وجدت ذدعة.

أكثر ، لدينا الكثير من الحمود. قد يكون هذا سيئاً بالنسبة لنا -أنا أنظر إليك ، مجموعة الصناديق "24" ومع ذلك ، يمكننا أيضاً تحويلها لصالحنا. اكتشفت أنه بمجرد أن بدأت في البرمجة وصنع الأشياء ، استوعبت كثيراً في المشروع لدرجة أنني لم أعد أهتم بالتلفاز أو الطعام أو النوم. كان هناك عدد غير قليل من عطلات نهاية الأسبوع عندما قمت بالتشغيل حتى شروق الشمس.

فكيف نستفيد من هذا القصور الذاتي؟

أولاً ، يجب أن تفهم أن تبديل المهام صعب للغاية. يتطلب الكثير من التحفيز. إذا كنت بمجرد وصولك إلى المنزل ، وجلست على الأريكة وشغلت التلفزيون ، فقد فقدت بالفعل ذلك المساء. هذا لأن مقدار الحافز المطلوب لتبديل المهام والقيام بشيء لا يحركه التطور مثل الأكل أو النوم هو مهمة شاقة.

هذا هو السبب في أن اللحظة التي تدخل فيها من الباب وتتحول إلى بيئة جديدة هي الأكثر أهمية. إذا أخبرت نفسك في هذه اللحظة أنك ستمارس 20 دقيقة من ممارسة البرمجة ، فمن المرجح أنك ستنتج وستستخدم جمودك في نهاية المطاف للتعلم لمدة ساعة أو أكثر. لن يدرك أي عقل مهمة تستغرق 20 دقيقة كمجهود كبير وينتهي بك الأمر بخداع عقلك للاستفادة من أمسياتك.

الخطوة التالية هي تطوير الحادة. تشير الأدب إلى أنه لتطوير عادة جديدة ، عليك القيام
بهذه المهمة يوميًا لمدة شهر. لقد استخدمت هذه الحيلة التالية لأحمال مختلفة

الأشياء ، من التمرين إلى الترميز ، فهي تعمل دائمًا مثل السحر. لتمهيد هذه الخدعة ، أريدك
أن تتخيل جدارًا به خمس لوحات معلقة ، أربعة منها متوازنة تمامًا ، أفقية تمامًا ، لكن
إحداها ملتوية.

الآن تخيل ذلك حقًا ، هل هناك جزء منك يريد إصلاحه؟

الآن دعنا نتخيل تقويًا شهريًا به مربعات تمثل أياها فردية. إذا قمت بتغذية هذه العادة
الجديدة في يوم معين ، فإنك تضع خطأ خلال ذلك اليوم.

إذا واصلت خطك في اليوم التالي ، فإنك تقوم بتمديد هذا الخط وما إلى ذلك. هناك شيء
ما يتعلق بعدم كسر الخط المستمر الذي يحفز معظم الناس على الاستمرار في تطوير عادة.
على الرغم من أن هذا يبدو غريبًا ، إلا أن هناك عدة مرات كنت سأستسلم فيها ، لكنني
اضطرت إلى الاستمرار بسبب خط طويل ومتواصل.

2

القاعدة الثانية

رمز الغرض



The
App Brewery

عندما بدأت في تعلم كيفية البرمجة لأول مرة ،

كانت هناك أوقات لا حصر لها عندما التقطتها ثم استسلمت مرارًا وتكرارًا. هذه قصة شائعة بين المبرمجين العصاميين. بالنظر إلى الوراء ، بعد تدريس الكثير من الطلاب ، أدركت أخيرًا ما يحدث. يبدأ الكثير من المبتدئين في تعلم البرمجة عن طريق اختيار لغة عشوائية واتباع مجموعة من البرامج التعليمية.

نسخ الكود ، سطرًا بسطر ، وأحيانًا كتابة التعليمات البرمجية لحساب الأعداد الأولية ، وفي أوقات أخرى للعثور على جميع الأرقام الزوجية.

ولكن هل تعلم؟ يمكنني العثور على الأعداد الأولية بشكل أسرع بكثير من خلال البحث في Google عنها ، ولا يعد اختيار الأرقام الزوجية أمرًا مثيرًا للاهتمام.

ها هي الحقيقة. إذا كنت تتعلم البرمجة من أجل تعلم البرمجة ، فسيكون من الصعب عليك أن تجيدها. المهارات التي تتطلب الكثير من الوقت لشحذها ، مثل البرمجة ، سوف تتغذى على مجموعة التحفيز الداخلي لديك. شيء من الداخل يجعلك تنسى الأكل والنوم. أستطيع أن أقول بصراحة أن البرمجة في مشاريعي الخاصة هي واحدة من أكثر الأشياء الممتعة التي أقوم بها.

فهو يجمع بين التفكير المنطقي والإبداع ، وفي النهاية تكون قد صنعت شيئاً ما. في معظم الحالات ، شيء لم يشهده العالم من قبل. شيء يمكن أن يجعل حياتك أسهل أو أكثر إمتاعاً. شيء يمكن أن يجعل الكثير من حياة الناس أسهل وأكثر إمتاعاً. إنه مثل صنع دراجة نارية مخصصة ورائعة في المرآب الخاص بك ، دون الحاجة إلى المرآب أو إنفاق سنت على المكونات.

هذا هو ما يحفز معظم الناس. الجزء الخلاق. صنع الجزء. لذا أحثك على البدء في تعلم البرمجة باتباع برنامج تعليمي يصنع شيئاً ، أي شيء.

بالطبع ، من غير المحتمل أن تتمكن في البداية من برمجة Clash of Clans أو Legends. League of الكنك ستكون قادرًا على صنع شيء مثير للاهتمام. يمكن أن تكون لعبة نرد أو تطبيق بطاقة فلاش. ولكن طالما أنك في نهاية البرنامج التعليمي ، ستكون قد صنعت شيئاً يمكنك استخدامه واللعب به ، فستكون متحمساً أكثر بكثير للتشجيع حتى النهاية.

خلال جميع دوراتنا ، نطلب دائماً من طلابنا التفكير في تطبيق بسيط يريدون إنشاؤه. شيء يستخدم المهارات التي تعلموها خلال الدورة ولكنه أيضاً سيمدهم قليلاً لأنه يتعين عليهم معرفة كيفية تضمين بعض الوظائف الجديدة.

كان لدينا طالب قام بإنشاء تطبيق يعمل

قبل دقيقة واحدة كل يوم لتسهيل الانتقال إلى وقت استيقاظ مبكر. هناك طالب قام بعمل دلف

تطبيق عرض شرائح مخصص كهدية لعيد الأم. أنشأ شخص آخر تطبيقًا يعمل كمؤقت لعمل شرائح اللحم لثالية بدءًا على وزنها وسمكها.

لا توجد حدود لخيالك. سيكون الأمر صعبًا عند بدء العمل على تطبيقك الخاص لأنه لا توجد إرشادات خطوة بخطوة ، ولكنه سيحدث أيضًا أكبر تحسن في قدرتك على الترميز.

سيمون سينك

3

القاعدة الثالثة

لا يوجد "مثالي"
لغة للتعلم



The
App Brewery

كلما أجريت محادثات كبيرة ،

سيكون هناك دائماً شخص واحد يسألني "أي لغة برمجة يجب أن أبدأ في التعلم أولاً؟"

هناك هذا التصور الشائع بأن هناك مكاناً ما توجد فيه لغة مثالية للمبرمجين المبتدئين. يجادل البعض بأنها بايثون ، والبعض يقول إنها جافا سكريبت.

لكنني أقول إنهم جميعاً على خطأ.

لغة البرمجة هي مجرد أداة. لا يختلف عن أي أداة أخرى في صندوق الأجهزة الخاص بك. إذا كنت ترغب في دق مسمار ، يجب أن تستخدم مطرقة. إذا كنت ترغب في إصلاح أنابيب المياه الخاصة بك ، فربما تحتاج إلى مفتاح ربط.

نعم ، من الممكن دق مسمار باستخدام جانب مفتاح البراغي ويمكن استخدام نفس لغة البرمجة لحل أنواع مختلفة من المشكلات. سيخبرك النجار أن أدواته المفضلة هي المطرقة وسيقول السباك إنها مفتاح البراغي ، لكنها ما زالت لا تجعلها "أفضل أداة لإصلاح الأشياء".

سيخبرك مطور الويب أن JavaScript هي أفضل لغة للتعلم للمبتدئين. سيخبرك خبير إنصائي بـ أنك ستحصل على أفضل خدمة بلغة البرمجة R. لكن في نهاية المطاف ، كل ما يهم هو ما

أنت تحاول أن تفعل بأداتك. إذا كنت تريد إنشاء تطبيقات ، iOS فتعلم Swift. إذا كنت ترغب في إنشاء مواقع الويب ، فستحتاج إلى JavaScript.

لكن الخبر السار هو أن مفاهيم البرمجة الأساسية: الحلقات ، الشرطية ، الوظائف ، إلخ. كلها متشابهة. الفرق هو في الغالب نحوي.

في اللغة الإنجليزية ، لدينا ذئاب ضارية ، ولدينا في ألمانيا Werwölfe. إنه لا يزال نفس الثدييات التي تمزق القميص والتي تظهر أثناء اكتمال القمر ، ولكن يتم تهجئتها بشكل مختلف.

الطباعة إلى وحدة التحكم في Swift: طباعة ("Hello Werewolves")

الطباعة إلى وحدة التحكم في Java: println ("Hello Werwölfe")

لذلك ، حدد المهمة التي تحاول إنجازها ، ثم اختر أفضل أداة لهذه المهمة.

4

القاعدة الرابعة

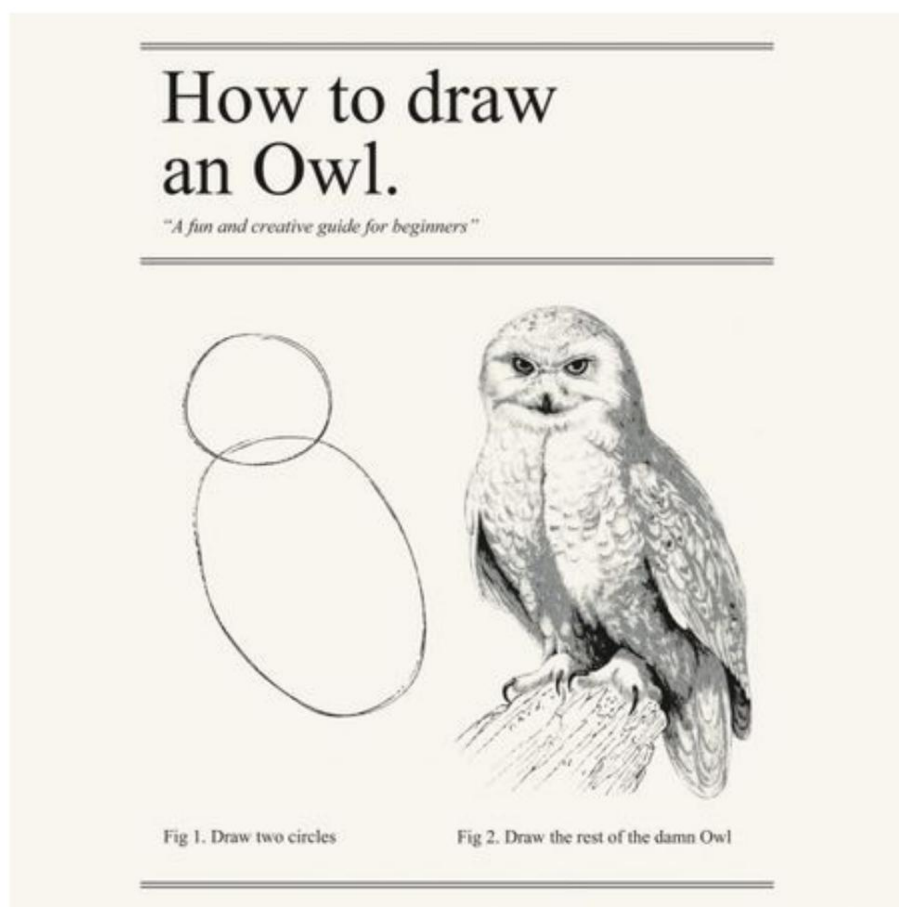
ماذا أفهم
أنت تكتب



The
App Brewery

لدي مشكلة في طريقة كتابة معظم دروس البرمجة.

هناك الكثير من البرامج التعليمية حيث ترى ظاهرة "هذه هي الطريقة التي ترسم بها البومة".



يبدو الأمر كما لو أن المبرمج لديه نوايا حسنة وبدأ بإظهار كيفية القيام بكل شيء خطوة بخطوة

ولكن بعد ذلك ، في مرحلة ما ، أدرك أنه شرّ في مهمة عبثية واستسلم. لقد شاهدت دروسًا حيث يبدأ المؤلف بمستوى رهيب من التفاصيل ثم يعود في منتصف الطريق إلى "أمن تمت ببساطة بإعداد قاعدة بيانات صحابه". مع الأخذ في الاعتبار أن هذا برنامج تعليمي موجه للمبتدئين!

هذا يؤدي إلى العديد من المشاكل. المشكلة الأكثر شيوعًا هي قيام الطالب بنسخ الكود الموجود في البرنامج التعليمي فقط وليس لديه أدنى فكرة عما يفعله أي منها. لماذا أضاف هذا السطر الإضافي بعد تحليل JSON؟ لماذا يصنع هذا القاموس بشكل مختلف عن السابق؟

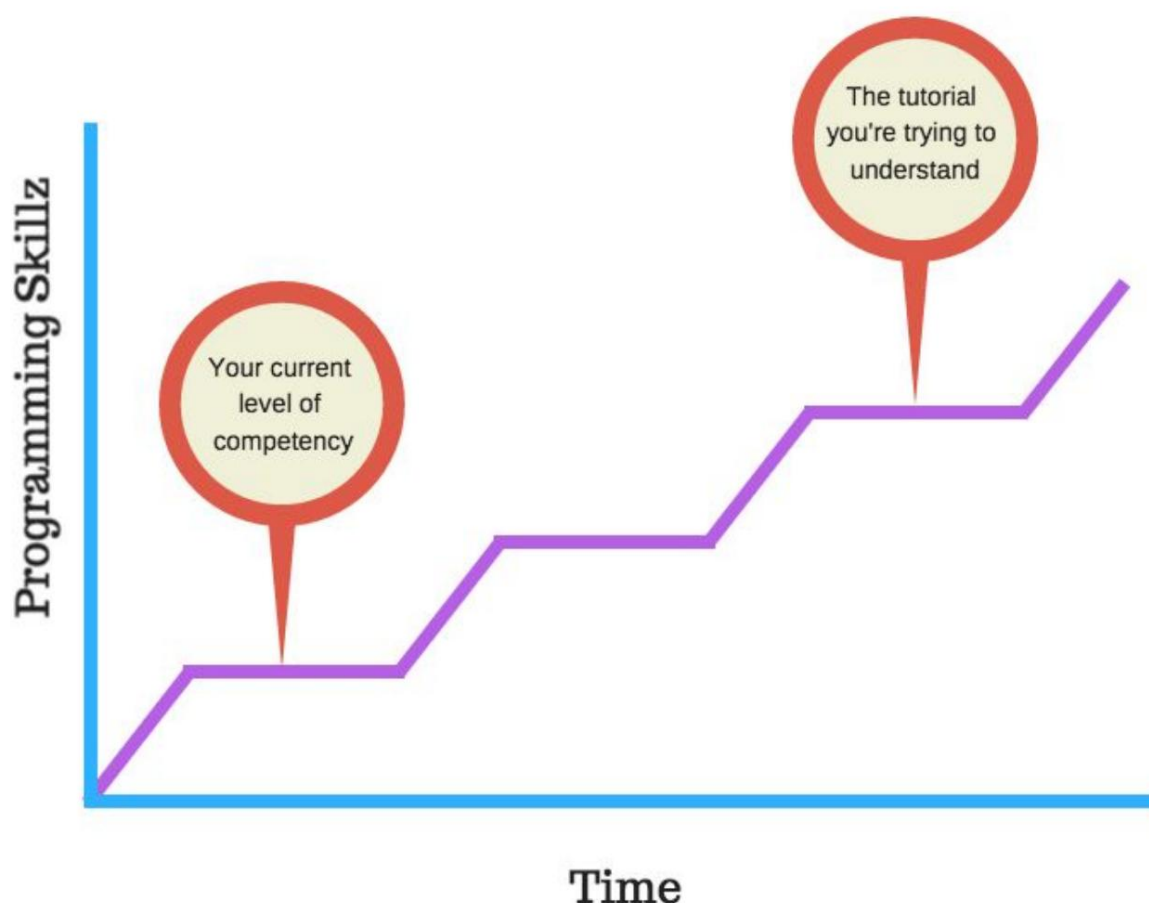
من السهل جدًا الحصول على ركبتيك عميقين في أحد هذه الأنواع من البرامج التعليمية لأنه يعد بتعليمك كيفية إنشاء "Flappy Bird" أو "Candy Crush" لكن في ثلثي المسافة ، لا شيء من الأشياء التي تكتبها منطقي وستبدأ في رؤية اللون الأحمر على الشاشة بالكامل. البق. الكثير منهم.

لماذا؟ لا يوجد فكرة. لا شيء يعمل. قضيت آخر 3 ساعات في نسخ الكود ولم تتعلم شيئًا سوى أن الترميز ربما يكون سيئًا.

لا تدخل في هذا الفخ. إذا رأيت برنامجًا تعليميًا ينتقل من المبتدئين إلى المتقدمين بعد السطر 3 أو يستخدم كلمة "ببساطة" بشكل متحرّج جدًا أو لا يشرح أيًا من التعليمات البرمجية ، فتوقف. اترك هذا البرنامج التعليمي.

هناك الكثير من الأسماك في البحر.

في أحيان أخرى ، يحاول المؤلف شرح ما يفعلونه. لكنك ما زلت لا تفهم شيئاً يقولونه ، فأنت في برنامج تعليمي متقدم لن يحسن البرمجة الخاصة بك. قد يكون من المغري بناء أشياء عظيمة ، خاصةً عندما تعد المدونة بإمكانية صنع أي شخص. ولكن إذا لم تتمكن من معرفة ما يحدث ، فمن الأفضل لك أن تبني أساساً أفضل.



مفتاح تعلم البرمجة هو كل شيء عن التعلية. تريد أن تتمدد مرارًا وتكرارًا وأن تُبنى المعرفة على المعرفة السابقة. إذا كان هذا المنحدر شديد الانحدار ، فسوف تضيق. إذا كان هذا المنحدر ضحلًا جدًا ، فسوف تشعر بالملل.

يختلف التدرج اللوني الصحيح من شخص لآخر. لهذا السبب نشجع الطلاب على استخدام وظيفة تغيير السرعة بحرية في برامجنا التعليمية.

بهذه الطريقة ، يمكنك الاستماع بسرعة مضاعفة إذا كنت مرتاحًا للمفاهيم والإبطاء إلى نصف السرعة إذا كان شيئًا غير مألوف وتحتاج إلى وقت لفهمه واستيعابه.

تحتاج إلى تعلم الطبخ بدونهم.

تيم فيريس

5

القاعدة الخامسة

لا بأس أن لا أعرف

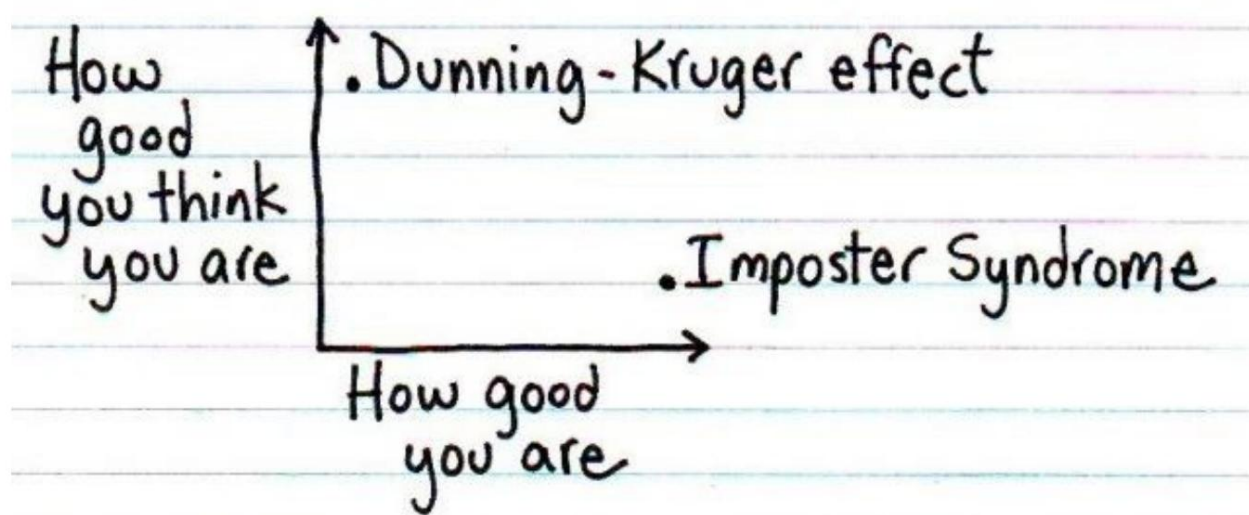


The
App Brewery

يُزعم أن مهندسي البرمجيات هم المهنة التي
تضم أكبر عدد من سكان Imposter

الذين يعانون من متلازمة

متلازمة Imposter هي ظاهرة نفسية يشعر فيها الناس وكأنهم محتالون ويقللون بشكل
كبير من مهاراتهم وقدراتهم.



يميل المبرمجون إلى النقد الذاتي ويشعرون باستمرار أن كل شخص آخر أفضل منهم في البرمجة.

إذا شعرت بهذه الطريقة من قبل ، فأنت لست وحدك ، كما تظهر الدراسات أن 70 بالمائة من الناس يعانون من متلازمة المحتال.

لقد رأيت مؤخرًا منشورًا على موقع الأسئلة والأجوبة Quora حيث سأل أحدهم: "هل سأطرد من Google (أو شركة تقنية كبيرة أخرى إذا تم القبض علي باستخدام StackOverflow كمرجع؟"

حصل على مجموعة من الإجابات الرائعة حقًا من المهندسين العاملين في Google و Amazon وشركات التكنولوجيا الكبرى الأخرى. سيخبرك أي شخص عمل كمهندس برمجيات أن عدم النظر إلى المراجع أمر مستاء للغاية. أتحداك أن تجد مبرمج Google واحدًا لم يستخدم Stack Overflow. (إذا لم تكن مألوفًا ، فإن Stack Overflow هو موقع أسئلة وأجوبة تعاوني للمبرمجين).

يخشى الكثير من المبرمجين الجدد أنه من خلال التحقق من المراجع وطلب المساعدة من الناس ، فإنهم سيظهرون أنفسهم على أنهم محتالون لا يعرفون البرمجة. لا أحد يستطيع الاحتفاظ بجميع المعلومات ذات الصلة في رأسه.

على سبيل المثال ، هذا هو اسم طريقة iOS:

```

- (const void *)dataBytes;
- (NSInteger)width;
- (NSInteger)height;
- (NSInteger)bitsPerPixel;
- (NSString *)colorSpaceName;
- (NSInteger)rowBytes;
- (NSInteger)pixelBits;

```

ما يقرب من 400 حرف!

في برمجة ، iOS هناك أكثر من 800 فئة و 9000 طريقة وهي في تزايد. في تطوير الويب ، هناك إطار عمل جديد كل أسبوع. لا أحد يتوقع منك أن تكون قادرًا على تذكر الرمز. هذا هو السبب الدقيق لكوننا مبرمجين ، يمكننا جعل الكمبيوتر يقوم بالأشياء المملة من أجلنا. على سبيل المثال ، رمز تسجيل الصوت هو مجرد بحث قصير ، فلماذا تحتاج إلى حفظه؟

المهارة التي يبحث عنها معظم أصحاب العمل عند التوظيف هي القدرة على التفكير. تُقدّر المعرفة في عالم يصعب فيه الحصول على المعلومات. في القرن التاسع عشر ، كان الأغنياء فقط هم من يستطيعون الوصول إلى الكتب الجيدة والمعلمين الجيدين. الآن ، كل شخص لديه كل المعلومات التي لديهم والمزيد بنقرة واحدة على الفأرة. المعلومات تفقد قيمتها ، والقدرة على التفكير هي الأسهم للشراء. لذلك لا تخف من البحث ، اسأل على StackOverflow أو ابحث عن موارد لمساعدتك في حل ملف

مشاكل. أفضل المبرمجين يفعلون ذلك.

المهارة التي تحتاج إلى صقلها هي في طرح أسئلة جيدة وفهم الإجابة. لا فائدة من نسخ ولصق التعليمات البرمجية من إجابة StackOverflow إذا لم يكن لديك أي فكرة عن كيفية عملها. نظرًا لأن StackOverflow يعمل على نظام سمعة ، فمن مصلحته أن يكون واضحًا قدر الإمكان في إجابته ليتم تمييزها على أنها صحيحة وجمع الأصوات المؤيدة.

في معظم الحالات ، ليس من المنطقي البدء في البحث في StackOverflow كلما واجهتك مشكلة. يجب أن يحاول الخيار الأول دائمًا اكتشافه بنفسك. لذا فإن برنامجك لا يفعل ما كنت تتوقعه ، ولكن قبل أن أكتب آخر ثلاثة أسطر من التعليمات البرمجية ، كان يعمل بشكل جيد. لذا دعنا نتعرف على ما حدث في هذه الأسطر الثلاثة الأخيرة الذي تسبب في كسر تطبيقي؟

إذا كنت لا تستطيع معرفة ذلك حقًا ، فابدأ بـ Google. ابحث عن استعلامك أو إذا كان لديك خطأ ، فقم بلصق رموز الخطأ ورسالة الخطأ. من المحتمل أن تكون مشكلات البرمجة لديك ، كمبتدئ ، شائعة جدًا ، وقد يستغرق شخص ما وقتًا لكتابة برنامج تعليمي واضح وموجز لمساعدتك على فهم الخطأ الذي تواجهه. مع ازدياد مهارتك في البرمجة ، تصبح المشاكل التي ستواجهها غامضة أكثر فأكثر ، ولكن نأمل ، إذا اتبعت القواعد الـ 11 الأخرى ، فستكون أيضًا مبرمجًا أكثر قدرة وتكتشفها بنفسك أو تعرف بالضبط من أين تحصل عليها يساعد.

السبب الآخر الذي يجعلك تبدأ مع Google هو أن خوارزمية البحث في StackOverflow تنظم الأسئلة والأجوبة حسب الحداثة وليس الشعبية. لقد تم طرح الكثير من المشكلات التي ستواجهها أثناء البدء والإجابة عليها منذ سنوات ، لكنها لا تزال تحظى بشعبية كبيرة.

لذا اسأل بحكمة وستجني الفوائد من المجتمع. في أحد الأيام عندما تصبح أنت خبيرًا في الكود ، سترد الجميل لنفس المجتمع وتساعد الجيل القادم من المبرمجين.

لا تخف من طلب المساعدة "

علامة قحتنا الجاهل طلب بالمهوا على القوية. إنه يظهر أن لديك الشجاعة للاعتراف عندما لا تعرف شيئًا ، وأن تتعلم شيئًا جديدًا.

باراك اوباما



القاعدة السادسة

كن مقلدا



The
App Brewery

في بداية رحلتي في البرمجة ،

اعتقدت أن طريقة تعلم البرمجة هي قراءة مجموعة كاملة من الكتب. اشتريت كتبًا على C++ و C# و Java وتحميل المزيد. سمها ما شئت ، كان لدي. لكنهم لم يفعلوا الكثير بخلاف جعلني في حيرة من أمري.

قرأت. وأبرزت. انا نسيت. غطت في النوم.

الكتب جيدة كمراجع. إذا كنت تريد التعمق في المندوبين والبروتوكولات ، فاقرأ الفصل الخاص بذلك. ولكن إذا كنت تريد أن تتعلم ، اصنع شيئًا.

لكن ماذا تصنع؟

تفتقر إلى الأفكار؟ كن مقلدا. اصنع المفكرة الخاصة بك ، وصنع MSPaint الخاص بك ، وصنع البيانو الخاص بك. إذا كنت من محبي الألعاب ، فاصنع كاسحة ألغام ، وصنع Tetris ، وصنع Flappy Bird. لن تكون مفيدة نوعًا ما فحسب ، بل ستكون أيضًا فرصة مثالية لك لمعرفة كيفية القيام بالأشياء واكتساب الخبرة في العثور على المساعدة. شيء جديد تمامًا في العالم مثل الهاتف الذكي ثلاثي الأبعاد

التوقعات ، لن يتمكن أحد من مساعدتك بها. من خلال إنشاء تطبيقات أو برامج مقلدة ، ستخطو في ملف الطريق الذي سلكه الكثيرون قبلك. بهذه الطريقة تزيد من الفرص التي يمكن أن يقدمها شخص ما يمكنك المساعدة والمشورة عندما تتعثر.

بابلو بيكاسو



القاعدة السابعة

كن مسؤولاً



The
App Brewery

كن مسؤولاً أمام شخص ما. اظهر عملك.

أكبر مشكلة في دورات البرمجة عبر الإنترنت هي الافتقار إلى المساءلة. لا شك أن هناك الكثير من الدورات التدريبية المفتوحة الضخمة على الإنترنت ، (MOOCs) مثل Coursera و Udacity و Udemy و Skillshare. ولكن ما هي عواقب عدم أداء واجبك المدرسي أو فقدان محاضرات لمدة شهر؟ لا شيء. لا أحد يهتم.

دعونا نواجه الأمر ، الدافع الداخلي ليس قوياً في أي منا. يمكننا دائماً العثور على سبب استحقاقنا لمشاهدة Netflix وتناول الآيس كريم. لا يمكنني حتى حساب عدد الدورات التدريبية عبر الإنترنت التي قمت بالتسجيل فيها وبالتالي لم أستمع إلى محاضرة واحدة أو أكملت جزءاً واحداً من الدورات الدراسية.

أنت بحاجة إلى المساءلة والالتزام بالتعلم. فكر في العودة إلى أيام دراستك الجامعية ، هل كنت ستزعج نفسك بإنهاء هذا المقال في الساعة 3 صباحاً إذا لم يكن هناك شيء يعتمد عليه؟ هل كنت ستذهب إلى أي من المحاضرات إذا كنت لا تهتم بالنجاح أو الفشل؟

هذا هو السبب في أننا نحاول إدخال المساءلة في دوراتنا. لقد أدركنا أن مطابقة الطلاب مع الأصدقاء تساعد. شخص آخر مبتدئ ، في نفس مستواك الذي يساعدك أحياناً وفي أوقات أخرى يحتاج إلى مساعدتك. في بعض الأحيان ، نظراً لتباين معدلات التعلم لدى الأشخاص أو إذا تم إقرانك بشخص كسول ، يمكنك تبديله والحصول على صديق جديد. نظراً لأن هذا النظام تطوعي تماماً ، فهناك درجة من الاختيار الذاتي للأشخاص الذين يعملون بشكل جيد في فرق ويتم تحفيزهم من قبل الآخرين. مثلما من المرجح أن تذهب إلى صالة الألعاب الرياضية إذا قمت بالتسجيل مع شريكك ، فمن المرجح أن تعرف ما إذا كان لديك رفيق تشفير.

لذلك إذا لم تكن في دورتنا ، فابحث عن دورتنا الخاصة. هناك الكثير من مجموعات Facebook المخصصة لأولئك الذين يتعلمون البرمجة. يوجد منتدى فرعي كامل (r / learnprogramming) مخصص لهذا ، أنا متأكد من أنك ستجد أشخاصاً متشابهين في التفكير في مكان ما عبر الإنترنت أو في وضع عدم الاتصال.

الشيء التالي الذي سأخبرك به سيكون مثيراً للجدل. نعتقد أن الناس لا يقدرّون الأشياء التي ليس لها قيمة. هذا هو السبب في أن كورسيرا تأخذ عددًا كبيراً من دوراتها المجانية. لقد رأوا أن ملايين الأشخاص كانوا يشتركون فيه ولكن لم يكن أحد يأخذ أيًا من الفصول الدراسية ، ناهيك عن إكمال أي من المشاريع. لقد كان في الواقع ضاراً بتعلم الطلاب لتقديم دورة تدريبية مجانية. لدينا جميعاً درجة من ميول الاكتناز ومن السهل جداً الاشتراك في مجموعة من الأشياء التي يمتلكها

المستقبل يمكنك أن تعاني من خلاله. تقول إن هناك دائمًا غدًا.

لذلك إذا كنت مدفوعًا بدافع خارجي أكثر منه داخليًا ، فحاول استخدام القليل من الحافز المالي لدفع تعلمك. فكر في قيمة إحدى المهارات الحياتية بالنسبة لك وضع أموالك حيث تكون نواياك. تعرف على ما إذا كنت تتعامل مع محتوى الدورة التدريبية بشكل أكبر مع الالتزام المالي أو بدونه. هناك الكثير من الأماكن حيث يمكنك الدفع مقابل شيء ميسور التكلفة لتحفيز نفسك على بدء عادة تعلم منتظمة.

الجزء الأخير من هذه القاعدة هو محاولة إيجاد طرق للتقييم. حسنًا ، لذا فإن الحصول على التقييم أمر مهم للغاية مع الموت والضرائب من حيث مدى تمتع الناس بها. ولكن عند تعلم أي شيء ، من المهم دائمًا الحصول على تعليقات. ستحصل على تقييم موضوعي لمستوى مهارتك الحالي ، بدلاً من الشعور بأنك محتال أو مليء بثقة زائفة. يوجد في Coursera نظام حيث يقوم الطلاب بتميز عمل بعضهم البعض. في ، App Brewery نستخدم تعليم Github لاختبار الكود الخاص بك والبحث عن الأخطاء والمشاكل المتعلقة بكودك. ولكن إذا كنت في دورة ترميز لا تحتوي على نظام مثل هذا ، فسيكون من المفيد لك العثور على موجه رمز يمكنه مراجعة التعليمات البرمجية الخاصة بك وتقديم ملاحظات لك. فقط ما يمكن قياسه يمكن تحسينه.



القاعدة الثامنة

استمر في التعلم



The
App Brewery

أن تكون مبرمجًا جيدًا يشبه إلى حد ما أن تكون مادونا.

لا تنفذ واشتري حمالات الصدر المخروطية حتى الآن. ما أعنيه هو أن البرمجة ستستمر في التطور. لكي تظل وثيق الصلة بالموضوع ، عليك أن تستمر في إعادة اكتشاف نفسك.

هناك دائمًا اتجاهات جديدة وتقنيات جديدة ولغات جديدة. يستمتع المبرمجون الرائعون بتعلم أشياء جديدة ، حتى لو كان ذلك يعني أنهم يصبحون مبتدئين مرة أخرى.

سيستمر العالم في التحرك ، إذا بقيت في مكان واحد ، فستترك في النهاية وراءك. أعرف مبرمجين لم يتعلموا أبدًا أي شيء آخر بخلاف فورتران. أعرف مبرمجي Objective-C الذين لا يستطيعون إقناع أنفسهم بالقيام بالقفزة وتعلم Swift ، على الرغم من أن Apple تخبر المطورين أنه سيتم التخلص التدريجي من Objective-C. نعلم جميعًا أن Apple لا تصدر أبدًا تهديدات لا تنفذها ، ما عليك سوى إلقاء نظرة على محرك الأقراص الضوئية (وقرصًا مقبس سماعة الرأس؟).

لا تكن محرك الأقراص الضوئية. أو بالأحرى ، لا تكن الكمبيوتر المحمول الذي لا يزال يحاول تشغيل الأقراص المضغوطة. إذا تغيرت احتياجاتك ، فتعلم

استخدم أداة جديدة. استمر في التعلم ، ابق على صلة.

هل أنت مطور ويب تريد دائمًا الدخول في تطوير الأجهزة المحمولة؟ اختر نظامًا أساسيًا وتعلم iOS أو Android. هل أنت مطور أمامي يغري بك المكس الكامل؟ التقط تطوير الويب باستخدام Node. إذا كنت تفهم بالفعل مفاهيم البرمجة الأساسية ، فسيكون اختيار المزيد من اللغات أسهل بكثير من البدء من الصفر.

يعد "Learn x in y minutes" مصدرًا رائعًا للمبرمجين الحاليين لتعلم لغات برمجة جديدة. تحقق من مواردهم هنا: learnxinyminutes.com

Learn X in Y minutes

[Share this page](#)

Select theme: [light](#) [dark](#)

Where X=Python

Get the code: [learnpython.py](#)

Python was created by Guido van Rossum in the early 90s. It is now one of the most popular languages in existence. I fell in love with Python for its syntactic clarity. It's basically executable pseudocode.

Note: This article applies to Python 3 specifically. Check out [here](#) if you want to learn the old Python 2.7

```
# Single line comments start with a number symbol.

""" Multiline strings can be written
    using three "s, and are often used
    as documentation.
"""

#####
## 1. Primitive Datatypes and Operators
#####
```



القاعدة التاسعة

لعب فووسبالل



The
App Brewery

عندما تشاهد افلام هوليود عن المبرمجين

عادة ما يجلسون أمام جهاز كمبيوتر محمول ، يخلطون لوحة المفاتيح كما لو كانوا في نوع من لعبة "سحق الخلد" عالية المخاطر.

عندما ترى المبرمجين الحقيقيين يعملون. تميل إلى أن تبدو مثل هذا:



نعم هذا صحيح. ممنوع الكتابة. مجرد التحديق. الكثير من التحديق.

في الشركة ، يميل الناس إلى الشكوى من أن المبرمجين يلعبون دائماً لعبة كرة القدم أو يفعلون شيئاً آخر لا يشبه العمل. قد لا يكون الناس قادرين على معرفة ذلك ، لكنهم في الواقع يعملون.

عندما تراهم يستمتعون بلعبة فووسبال ، يضحكون ويمزحون ، فمن المحتمل أنهم يعانون من الداخل. لوجود خطأ ، هناك دائماً خطأ. أو هناك شيء غامض بشأن الكود الخاص بهم لا يمكنهم حله. ربما تعمل الشفرة بشكل مثالي ، ولكن بشكل غير متوقع (لا يحب المبرمجون أي شيء غير متوقع بالمناسبة). كما لو أنهم كتبوا ألف سطر دفعة واحدة وبشكل غير متوقع لا توجد أخطاء.

**I Am Devloper**

@iamdevloper

Follow



before bed: drink peppermint tea, run a bath, meditate, get into bed, read, fill out your gratitude journal, turn the lights out, roll over...

brain: hi, that bug you couldn't fix earlier was on L34 in helpers.js, bye

🌿_🌿

2:32 am - 21 Nov 2018

1,968 Retweets 8,289 Likes



78

2.0K

8.3K

قد لا يفهم الآخرون ، ولكن في هذه المواقف ، من المفيد دائماً الابتعاد عن التعليمات البرمجية الخاصة بك وإعطائها بعض الوقت والمسافة.

هل لديك خطأ في التعليمات البرمجية الخاصة بك لا يمكنك حلها؟
نم عليها ، لعب كرة القدم ، اذهب في نزهة على الأقدام. في 9 من أصل 10 حالات ، سيصبح الحل واضحاً. في حالة واحدة من أصل 10 حالات متبقية ، تكون قد انتهكت للتو.

قد يبدو هذا غير بديهي ، لكن نصيحتي هي دائماً تقليل الترميز والتفكير أكثر. بمجرد كتابة التعليمات البرمجية غير المدروسة وإحضارها إلى العالم ، سيتعين عليك حتماً العودة وتمشييط التعليمات البرمجية الخاصة بك ، سطرًا بسطر ، وإعادة هيكلة وحذف الأشياء. هذه دائماً تجربة مؤلمة.

لذا تذكر أن أسهل رمز للتخلص منه هو الكود الذي لم يُكتب مطلقاً.



حكم عشرة

احصل على مرشد



The
App Brewery

عندما كنت أتعلم الفرنسية ،

صادفت طريقة أدت إلى أكبر قفزة في قدرات التحدث لدي. كان هناك تبادل لغوي عبر سكايب. كنت أقوم بالاشتراك مع متحدث فرنسي أصلي يريد تعلم اللغة الإنجليزية. كنا نقضي نصف ساعة في التحدث بالفرنسية ونصف ساعة في التحدث باللغة الإنجليزية. كنا نخصص ساعة كل أسبوع لتحسين اللغة التي كنا نحاول تعلمها.

بينما كنا نجري محادثة باللغة الفرنسية ، كان يصحح لفظي أو نحوي ويقترح طرقًا يمكنني من خلالها بناء جملتي لتبدو أكثر أصالة.

البرمجة الزوجية هي تقنية تطوير برمجيات رشيقة تعتمد على مبادئ متشابهة جدًا. على سبيل المثال ، يجلس المتعلم والموجه في نفس محطة العمل ويعملان على حل مشكلة ما. المتعلم هو المسؤول عن كتابة الكود والموجه يراجع الكود سطرًا بسطر كما هو مكتوب.

قد يكون الأمر غير مريح في البداية لأنه قليل

احراج ارتكاب الاخطاء وجعلها تشير لك. ولكن إذا كان لديك معلم جيد ، فسيقدم لك عقودًا من الحكمة المتراكمة التي يمكن أن تؤدي إلى تحسينات هائلة في قدرتك ، كل ذلك في غضون ساعات قليلة.

ستتمكن من الوصول إلى شخص كان لديه الوقت لصقل مهاراته ، وإيجاد طرق فعالة للقيام بالأشياء وإظهار كيفية قيامه ببرمجة المشكلات والتعامل معها.

لا يحل الموجهون الجيدون مشاكلك ، بل يمارسون الطريقة السقراطية في طرح الأسئلة الجيدة التي تجعلك تفكر بنفسك. إذا سألتني عن كيفية كتابة مكالمة عبر الشبكة ، فبالطبع يمكنني ببساطة كتابتها بالكامل وإقناعك بنسخها. لكن هذا لا يساعدك. بدلاً من ذلك ، إذا أوضحت لي كيف تتعامل مع المشكلة وأوضحت لك كيف أتعامل مع المشكلة ، فيمكنك أن تتعلم أكثر بكثير من مجرد اتباع الوصفة.

في المرة التالية التي تواجه فيها مشكلة مختلفة ، يمكنك تطبيق نفس النهج والبدء في حلها بنفسك.

تذكر دائمًا أن المعلومات رخيصة. قبل قرن من الزمان ، إذا كنت أرغب في التعرف على أسباب المرض ، فربما كان علي أن أكون أرستقراطيًا ، أو أقطع الحطب وأحمل الماء للسيد وأصبح متدربًا. في الوقت الحاضر يمكنني البحث في Google والحصول على إجابتي في بضع ثوانٍ.

لذلك لا تتوقف عن المعلومات. تعلم أن تفكر بدلاً من ذلك. كيفية التعامل مع المشكلة. كيفية حل المشكلة. كيفية تأطير المشكلة. ستأخذك هذه المهارات إلى أبعد من مجرد الحفظ والقلس.

لكن أين تجد مرشدًا؟

هناك لقاءات متعلقة بالبرمجة تحدث في كل مدينة تقريبًا في العالم. اذهب إلى www.meetup.com والعثور على واحدة ذات صلة بلغة تحاول تعلمها. [حضور اللقاءات](#) والتعرف على الناس. تبادل خبراتك لخبراتهم. ربما شخص ما يحتاج إلى محاسب ، ربما شخص ما يحتاج إلى استشارة قانونية. استبدل وقتك بوقتهم. لا تقل لشخص ما ، "هل ستكون مرشدي؟". لا أحد يريد التخلص من وقت فراغه لبعض الغرباء. بدلاً من ذلك ، اعرض مساعدتك مقابل مساعدتهم وستنجح في العثور على مرشد 95% من الوقت.

11

القاعدة الحادي عشر

ادخل في هذه العادة تقطيع



The
App Brewery

إذن لديك فكرة رائعة.

لكنها طريقة معقدة للغاية بالنسبة لمستوى مهارتك الحالي. ماذا تفعل؟ تنضم إلى Express. Chunking

كلا ، أنا لا أتحدث عن فيلم بيت الفن. أنا أتحدث عن حل مشكلة البرمجة الخاصة بك.

لنفترض أنك تحاول صنع روبوت يمكنه صنع خبز محمص بالزبدة. (إذا كان أي شخص يعمل على أحد هذه الأشياء ، فسأكون سعيدًا بتمويل (Kickstarter!) لا يعرف الروبوت أي شيء عن الخبز المحمص أو الزبدة أو السكاكين. صدق أو لا تصدق ، يتطلب الأمر في الواقع دوائر معقدة جدًا في أدمغتنا لنكون قادرين على تحقيق شيء بسيط مثل الزبدة على شريحة من الخبز المحمص.

(ربما هذا هو السبب في أنني لا أستطيع أن أفعل ذلك بدون قهوة).

لذا فإن إنشاء روبوت يقوم بكل ذلك بشكل مستقل أمر معقد وصعب حقًا. ولكن نظرًا لأننا مبرمجون جيدون ، يمكننا إجراء بعض التقسيم وحل المشكلة.

لا يحتاج الروبوت حقًا إلى معرفة ما هو الخبز المحمص و

ما هي الزبدة ، نحن لا نصنع Skynet هنا ، لذلك دعونا نتمسك بالأشياء العملية. هناك ثلاثة أشياء نحتاج إلى الروبوت للقيام بها:

التقط وترتيب قطعة التوست في وضع الزبد المثالي.

التقط حصة من الزبدة.

ضع الزبدة على الخبز المحمص مع تغطية جيدة (هذا هو الجزء الذي أجده أصعب).

بعد ذلك ، تقوم بتقسيم كل وحدة إلى أبعد من ذلك. في هذه العملية ، يمكنك التفكير في طرق بديلة لحل المشكلة. على سبيل المثال ، هل يحتاج الروبوت إلى "دهن" الزبدة؟ أو يمكن فقط إذابة الزبدة على الخبز المحمص؟ هل تحتاج إلى تعلم كيفية التقاط السكين؟ أو هل يمكن أن يكون لها نوع من ذراع السكين يحمل في ثناياه عوامل ، مثل نوع من قرصنة السجن؟

كلما قمت بتفصيل المشكلات وتحديد المشكلة التي تحاول حلها ، كان من الأسهل تجميع التعليمات البرمجية الخاصة بك في أجزاء صغيرة الحجم. كلما كانت القطعة أبسط ، كان من الأسهل معالجتها.

لذا في المرة القادمة التي تحاول فيها إجراء ذلك "التقاطع بين Snapchat و Evernote" ، تذكر تقسيم المشكلة إلى أجزاء قابلة للحل.

12

القاعدة الثاني عشر

كسر رمز شخص آخر



The
App Brewery

واحدة من أهم الخطوات التي يجب اتخاذها

من أجل الانتقال من المبرمج المتعلم إلى المبرمج الكامل ، فهم كيفية الحصول على المساعدة. الجميع بحاجة إلى المساعدة. الجميع ، بما في ذلك من يسمون "مبرمجي مستوى الله".

لكن ما تفعله بالمساعدة سيحدد مدى سرعة تقدمك كمبرمج. على موقع مثل StackOverflow ، قد يكون من المغري للغاية نسخ ولصق الكود الذي قدمه شخص ما. يعمل برنامجك تمامًا كما كنت تأمله ، وتذهب في طرق البرمجة المرحلة.

لم يعلمك هذا التمرين أي شيء آخر غير الاعتماد على الكود. لأنه في المرة التالية التي تواجه فيها نفس المشكلة ولكن في موقف مختلف ، قد لا يعمل مقتطف الشفرة نفسه الذي قدمه شخص ما بعد الآن.

ثم ماذا تفعل؟ انت عالق.

هذا هو السبب في وجود قاعدة في البرمجة تنص على "عدم نسخ ولصق الكود الذي لا تفهمه". إذن ما الذي يجب عليك فعله عندما تواجه كتلة من التعليمات البرمجية التي تحل مشكلتك ولكن ليس لديك أدنى فكرة عن كيفية عملها؟ كسرهما.

الخطوة - 1 انسخ الكود والصقه في برنامجك. (نعم ، نعم ، أعلم أنني قلت للتو لا أفعل ذلك ، الصبر ، الصبر).

الخطوة - 2 تأكد من أن البرنامج أو التطبيق الخاص بك يعمل كما هو متوقع. أيؤكد أن كتلة الشفرة قد حلت مشكلتك حقًا.

الخطوة - 3 حذف النسخة ولصقها من الكود سطرًا سطرًا.

الخطوة - 4 في كل مرة تقوم فيها بحذف سطر ، تحقق لمعرفة ما تم كسره. هل التطبيق لا يزال يعمل؟ ما هي رموز الخطأ؟ ما الذي فعله حذف هذا السطر من التعليمات البرمجية لبرنامجك؟

الخطوة - 5 حتى إذا كنت تعتقد أنك تعرف ما يفعله سطر التعليمات البرمجية ، فاحذفه بأي طريقة. أهم مهمة كمبرمج هي اختبار افتراضاتك دائمًا مقابل النتيجة. أكثر شعور ممتع كمبرمج هو أن يتحقق العالم الحقيقي من صحة افتراضاتك. هل تعلم كم هو لطيف عندما يقول صديقك / صديقتك / زوجك / زوجتك تلك الكلمات الثلاث السحرية؟

"كنت على حق".

إنه من هذا القبيل ، لكن أفضل.

الخطوة - 6 قم بتبديل بعض الخطوط حولها. هل يمكن تحقيق نفس الوظيفة بترتيب مختلف للخطوط؟

لماذا كُتبت بالترتيب الذي كُتبت به؟

من خلال كسر رمز الحل ، سطرًا بسطر ، ستتعلم وتفهم ما يفعله كل سطر ولماذا تمت كتابته.

هذه طريقة أفضل بكثير لاستخدام التعليمات البرمجية من أشخاص آخرين من مجرد لصقها والأمل في الأفضل. بمجرد أن تفهم سبب أهمية كل سطر من هذه الأسطر ، في المرة التالية التي تواجه فيها مشكلة مماثلة ، ستتمكن من حل المشكلة بنفسك.

بمجرد إتقان الكود المتقطع من StackOverflow ، فإن المورد التالي الذي يجب استهدافه هو GitHub. إنها أداة يستخدمها المبرمجون للتعاون ولكنها أيضًا واحدة من أكبر مستودعات الكود مفتوح المصدر.

فكيف يمكنك استخدامه لتصبح مبرمجًا أفضل؟
لنفترض أنك تريد استنساخ Instagram. لكن لسوء الحظ ، لا تعرف كيف تفعل ذلك. لذا توجه إلى github.com وابحث عن "Instagram" أو "تطبيق الصور".

حتمًا ، سيكون هناك شيء مكتوب بلغة Java / Objective-C / Swift يمكنك تنزيله وإلقاء نظرة عليه.

فكر في هيكل برنامجهم. ألق نظرة على

كل الطبقات ، الثوابت ، التفاعل. قم بإجراء بعض التعديلات على الكود. هل ما زالت تعمل أم أنك كسرتها؟ لماذا كسرتها؟ هل هناك ارتباط لم تحدده؟ ادأل نفسك مجموعة من الأسئلة ، وتعلم من خلال الطريقة السقراطية. اعدم المشروع وافهم كيف تم بناؤه.

عندما تبدأ في أن تصبح جيدًا حقًا في هذا ، فإن الشيء التالي الذي يمكنك تجربته هو الهندسة العكسية. ابحث عن مشروع صغير على GitHub بواسطة مبرمج حسن السمعة ، قم بتنزيل التطبيق. قم بتشغيله وشاهد كل وظائفه. العب معها.

ثم قم ببنائه من البداية وبمجرد الانتهاء ، قارن الكود الخاص بك مع الكود الخاص بهم. هل هناك مكاسب في الكفاءة كان من الممكن أن تحققها؟ هل توجد حلول لأشياء لا يمكنك اكتشافها؟ أنت الآن تشارك حقًا في بطولات الدوري الكبرى.

مارك زوكربيرج

إبدأ اليوم

تعلم البرمجة.

أنت مشغول وتعمل بجد ، فأنت تستحق الوصول إلى أفضل دورات البرمجة في العالم. مع أكثر من مليون تقييم 5 نجوم ، يمكنك الوثوق في App Brewery.

يتعلم أكثر



The
App Brewery