eklektika
Language Means Business

Programming and development

```
fun append (xs, ys) =
    if null xs
    then ys
    else (hd xs):: append (tl xs, ys)

fun map (f, xs) =
    case xs of
        [] => []
      | x :: xs' => (f x)::(map (f, xs'))

val a = map (increment, [4,8,12,16])
val b = map (hd, [[8,6],[7,5],[3,0,9]])
```

Level B1

1. Advice for programmers

Reading / grammar. Read the following fragment (www.sitepoint.com) and correct six errors.

The best coders go through several phases on there programming journey:

1. **The "I know nothing" phase**
   Everything is new, nothing isn't easy.
2. **The "it's starting to make sense" phase**
   You've written few programs and are making fewer mistakes.
3. **The "I'm invincible" phase**
   Your confidence is as big as your competence. No challenge seems to difficult.
4. **The "I know nothing" phase, part II**
   You suddenly realize that development is much more complex and you begin to doubt you own abilities.
5. **The "I know a bit and that's OK" phase**
   You have decent coding skills but recognize your limitations and can find solutions to the most problems (even if that means hiring other developer).

Discussion. How does the above fragment relate to your own experience?

Before you read. What advice would you give to beginning programmers?

Can you think of any advice related specifically to the following: memory, networks, compilers, complexity, libraries?

7 timeless lessons of programming 'graybeards'

www.infoworld.com

By Peter Wayner InfoWorld March 9, 2015

The software industry loves the young. If you have a family, you're too old to code. Unfortunately, the young aren't always the best solution. While they know all the details of the latest, trendiest architectures and frameworks they lack fundamental experience with how software really works and doesn't. This experience comes only after many weeks of frustration over bugs.

In the spirit of sharing, here are several lessons that can't be learned by following the latest hype for a few weeks.

1. _____

Not so long ago, computer RAM was measured in megabytes, not gigabytes. Those who remember that know that memory should be treated like gold. Today, kids allocate RAM left and right. They leave pointers dangling and don't clean up their data structures because memory seems cheap. They know they click on a button and the hypervisor adds another 16GB to the cloud instance.

Then there's the danger of virtual memory. It is great in theory, but slow in practice. Software that runs fast in development may be very slow when many people use it.

2. _____

The people who sell the Cloud want you to believe that it is a simple web service that provides permanent, backed-up storage that you will never have to worry about. But even if you don't have to worry, you will have to wait for it. All traffic in and out of computers takes time. Computer networks are much slower than the traffic between the CPU and the local disk drive.

Programming graybeards grew up in a time when sending data took ages. This taught them that the right solution is to do as much as you can locally and write to a distant Web service only when everything is as small and final as possible.

3. _____

When software doesn't work, we assume that the problem must be in our code. We forgot to initialize something, or we forgot to check for a null pointer. In fact, however, many of the most annoying errors aren't our fault. Instead, they are the fault of the compiler or the interpreter. Old programmers learned long ago that sometimes the best way to debug an issue is to test our tools, not our code.

4. _____

Usability studies show that people's minds start to wander after 100 milliseconds. The constructors of old mainframe computers knew this. Sure, the machines could crash, but when they ran smoothly, they were quick.

If your project is AJAX-heavy, with too many JavaScript libraries and too much data flowing to the browser, it will be slow and users will hate it. Even if it looks cool and has better graphics.

5. _____

Modern websites can be a real test for the impatient. It can often take several seconds for the megabytes of JavaScript libraries to arrive. Then the browser has to push these multilayered megabytes through a JIT compiler. If we added up all of the time the world spends recompiling jQuery, it would be thousands or even millions of years.

This is an easy mistake for programmers who are in love with browser-based tools that use AJAX everywhere. It all looks great in the demo at the office. After all, the server is usually on the desk back in the same building. But the users on a DSL line? They're still waiting for the libraries to arrive.

6. _____

On one project, we asked a youngster who knew Greasemonkey backward and forward for help. He rewrote our code and he certainly made it more elegant. But the algorithmic complexity went from O(n) to O(n^2). He was sticking data in a list in order to match things. It looked pretty, but it would get very slow as n got large.

He didn't seem to understand why we weren't happy. We replaced his list with a hash table and all was well again. He's probably old enough to understand by now.

7. _____

The people who write libraries often build a Swiss-army knife that can handle many different versions of the problem and not something optimized for the problem that you have. That's good engineering and great coding, but it can be slow.

I remember a young programmer who laughed at my code because I wrote 10 lines to pick characters out of a string. "I can do that with a regular expression and one line of code," he said. "Ten-to-one improvement." He didn't think that his one line of code would parse and reparse that regular expression every single time it was called.

Libraries and APIs can be great when used well. But if they're used in the inner loops, they have a devastating effect on speed.

Comprehension / discussion. For each piece of advice, explain what, in the author's view, the difference between young and old programmers is. Do you agree with him? Do you think his advice is valuable?

Writing. Write a short, concise headline for each of the seven pieces of advice discussed in the article.

Vocabulary. Below are the definitions of some of the technical terms used in the text. Write them out without consulting the text, if possible.

_____ The overall design of a computing system and the logical and physical relationships between its components. It specifies the hardware, software, access methods and protocols used throughout the system. (Intro)

_____ A universal, reusable software environment that provides particular functionality as part of a larger software platform to facilitate development of software applications, products and solutions. (Intro)

_____ A coding error. (Intro)

_____ A pointer that does not point to a valid object of the appropriate type. (1)

_____ A piece of computer software, firmware or hardware that creates and runs virtual machines. (1)

_____ Virtual environment in the cloud. (1)

_____ Communication. (2)

_____ On the machine one is working on. (2)

_____ To assign an initial value to a data object or variable. (3)

_____ A pointer that does not point to any object or function. (3)

_____ A program that translates source code from a high-level programming language to a lower level language (e.g., assembly language or machine code). (3)

_____ A computer program that directly executes, i.e. performs, instructions written in a programming or scripting language, without previously compiling them into a machine language program. (3)

_____ To remove coding errors. (3)

_____ The degree to which a piece of software can be used by specified consumers to achieve objectives related to effectiveness, efficiency, and satisfaction. (4)

_____ To stop working. (4)

_____ A collection of non-volatile resources used by computer programs, often to develop software. (4)

_____ With many layers. (5)

_____ Compiler that works when the program runs. (5)

_____ Cross-platform JavaScript library. (5)

_____ A structure that maps keys to values. (6)

_____ A sequence of characters that define a search pattern. (7)

_____ To analyze a string of symbols. (7)

_____ A loop nested within another loop. (7)

Grammar. Without consulting the text, fill the gaps with the most appropriate verb form.

- If you _____ a family, you _____ too old to code.
- Even if you _____ to worry, you _____ to wait.
- If your project _____ AJAX heavy, it _____ slow and users will hate it.
- Even if it _____ cool and has better graphics.

- If we _____ all of the time the world spends on recompiling jQuery, it _____ thousands or even millions of years.

- If they _____ in the inner loops, they _____ a devastating effect on speed.

Kite – your programming copilot

https://www.youtube.com/watch?v=YkXzAbO2sHg

Comprehension. True or false:

- The explosion of new languages and libraries is a bad thing
- Completions are listed alphabetically
- The speaker doesn't like 'Batman vs. Superman'
- Kite requires an internet connection
- Kite does not include a search function

Close listening. Listen to the video again and fill the gaps in the phrases used by the speaker:

- packages that start with this _____
- instant _____ times
- dictionary with _____ parameters
- a one-click _____
- missing _____ statement
- constant arguments with the _____
- bring your own _____
- _____ privacy

Discussion. Consider the following questions:

- What do you think about this presentation?
- Would you find Kite useful in your work?
- Do you agree that in the future all programmers will be using such cloud-powered assistants?

Before you watch 1. There are many programming languages, with new ones appearing all the time. What should a programmer take into consideration when choosing which ones to learn?

Before you watch 2. Below are some words you will hear in the video. Match them with definitions provided.

fade away	thrive	social security	embrace	trumpet
struggle	catch on	demand	outlier	
widespread	transition	ample	momentum	

- to disappear gradually
- change
- something atypical or unexpected
- to adapt
- to do with difficulty
- used by many
- US government-funded retirement system
- to become popular
- to function successfully
- numerous, many
- to declare with enthusiasm
- increasing popularity

Which languages will stand the test of time?

https://www.youtube.com/watch?v=qz8zllnxjvc

Comprehension. Answer the following questions:

- What criteria for choosing languages to learn are mentioned and which are the important ones?
- In what context does the speaker talk about *inertia*?
- What is the best opportunity for a new language?
- What are the potential future 'outliers' for programmers to watch?

Discussion. What do you think of the speaker's arguments?

2. Agile and Scrum

Vocabulary review 1. Fill the gaps in the following description from www.scrumalliance.org. Sometimes you will have to combine two words, and sometimes you may have to change the form of the word. Some words may be used more than once.

sprint	plan	backlog	retrospective	ship
master	scrum	framework	product	review
own	stakeholder	day	customer	

The Scrum _____ in 30 seconds

- A _____ creates a prioritized wish list called a _____.
- During _____, the team takes a small chunk from the top of that wish list, *a* _____, and decides how to implement those pieces.
- The team has a certain amount of time — a _____ (usually two to four weeks) — to complete its work, but it meets each day to assess its progress (_____).
- Along the way, the _____ keeps the team focused on its goal.
- At the end of the _____, the work should be potentially _____: ready to hand to a _____, put on a store shelf, or show to a _____.
- The _____ ends with a _____ and _____.
- As the next _____ begins, the team chooses another chunk of the _____ and begins working again.

Vocabulary review 2. Fill the gaps in this fragment from Wikipedia, putting the verb in the correct form. Some may be used more than one and some may not be needed at all.

educate	organize	ask	define	announce
assure	serve	ensure	demonstrate	
steer	negotiate	communicate		

The following are some of the communication tasks of the product owner to the stakeholders:

- _____ the solution to key stakeholders who were not present at a sprint review;
- _____ and _____ releases;

- _____ team status;
- _____ milestone reviews;
- _____ stakeholders in the development process;
- _____ priorities, scope, funding, and schedule;
- _____ that the product backlog is visible, transparent, and clear.

Grammar. Write out the full questions to complete the following fragments from www.scrumalliance.com.

A good agenda for the scrum of scrums meetings is similar to the standard agenda for the daily scrum. In that meeting each team member is asked:

- you / do / yesterday?

  _____

- you / do / today?

  _____

- obstacles / slow down / you?

  _____

Because the scrum of scrums meetings may not be daily and because one person is there representing his or her entire team, these three questions need to be rephrased a bit:

- your team / do / since / we / last / meet?

  _____

- your team / do / before / we / meet / again?

  _____

- anything / slow down / your team?

  _____

The sprint retrospective includes three main questions/points for discussion:

- go well / during / the sprint cycle?

  _____

- go wrong / during the sprint cycle?

  _____

- we / can do / differently / to improve?

_____

Scrum vs. Kanban

https://www.youtube.com/watch?v=9Jgu1BlTlSc

Comprehension. Answer the following questions:

- What happened in 2012?
- What happened in 2013?
- What was the difference between the two teams?
- What do you think – how did this experiment end?