wiki Documentation

Release

Yucheng Wu

Apr 22, 2018

Contents

Research

[1]

---

[1] https://ying-zhang.github.io/misc/2017/ccf-all-in-one/

Linux

g[1]

---
[1] Linux () http://book.douban.com/subject/2208530/

Latex

3.1 Latex troubleshooting

> **Warning:** ! LaTeX Error: File 'string.sty' not found.
>
> Type X to quit or <RETURN> to proceed, or enter new name. (Default extension: sty)
>
> Enter file name:

nice   scientific   pictures   show   off.   ..   https://tex.stackexchange.com/questions/34482/
contest-show-off-your-skillz-in-tex-friends-tex-sx-birthday

The Comprehensive LATEX Symbol List.

http://melusine.eu.org/syracuse/

Browse All graphics generated with PGFPlots

https://www.tug.org/interest.html#latexdoc

CHAPTER 4

Kubernetes

Cluster-level Logging of Containers with Containers[1][3][4]

Borg, Omega, and Kubernetes[2]

---

[1] https://queue.acm.org/detail.cfm?id=2965647
[3] https://dl.acm.org/citation.cfm?id=2965647
[4] https://docs.fluentd.org/v0.12/articles/docker-logging-efk-compose
[2] https://queue.acm.org/detail.cfm?id=2898444

English

5.1 Abstract

Guidelines for adopting frontend architectures and patterns in microservices-based systems Microservice-based systems enable the independent development, deployment, and scalability for separate system components of enterprise applications. A significant aspect during development is the microservice integration in frontends of web, mobile, and desktop applications. One challenge here is the selection of an adequate frontend architecture as well as suitable patterns that satisfy the application requirements. This paper analyses available strategies for organizing and implementing microservices frontends. These approaches are then evaluated based on a quality model and various prototypes of the same application implemented using the distinct approaches. The results of this analysis are generalized to a guideline that supports the selection of a suitable architecture.

Improving performance of automatic program repair using learned heuristics Automatic program repair offers the promise of significant reduction in debugging time, but still faces challenges in making the process efficient, accurate, and generalizable enough for practical application. Recent efforts such as Prophet demonstrate that machine learning can be used to develop heuristics about which patches are likely to be correct, reducing overfitting problems and improving speed of repair. SearchRepair takes a different approach to accuracy, using blocks of human-written code as patches to better constrain repairs and avoid overfitting. This project combines Prophetâs learning techniques with SearchRepairâs larger block size to create a method that is both fast and accurate, leading to higher-quality repairs. We propose a novel first-pass filter to substantially reduce the number of candidate patches in SearchRepair and demonstrate 85% reduction in runtime over standard SearchRepair on the IntroClass dataset.

Repleo: a syntax-safe template engine - https://dl.acm.org/citation.cfm?id=1289977 Templates are a very common solution to generate code. They are used for different tasks like rendering webpages, creating Java Beans and so on. Most template systems have no notion of the object language and just generate text. The drawback of this approach is the possibility to generate syntactical incorrect code. This can lead to all kinds of annoying errors.

In this paper we present an approach for a syntax safe template engine. Syntax safety guarantees that the generated code can be correctly parsed. To ensure this we use the object language grammar to evaluate the template.

Detecting overly strong preconditions in refactoring engines Refactoring engines may have overly strong preconditions preventing developers from applying useful transformations. We find that 32% of the Eclipse and JRRT test suites are concerned with detecting overly strong preconditions. **In general, developers manually write test cases, which is costly and error prone.** Our previous technique detects overly strong preconditions using differential testing. However, it needs at least two refactoring engines. In this work, we propose a technique to detect overly strong preconditions in refactoring engines without needing reference implementations. We automatically generate programs and attempt to refactor them. For each rejected transformation, we attempt to apply it again after disabling the preconditions that lead the refactoring engine to reject the transformation. If it applies a behavior preserving transformation, we consider the disabled preconditions overly strong. We evaluate 10 refactorings of Eclipse and JRRT by generating 154,040 programs. We find 15 overly strong preconditions in Eclipse and 15 in JRRT. Our technique detects 11 bugs that our previous technique cannot detect while missing 5 bugs. We evaluate the technique by replacing the programs generated by JDOLLY with the input programs of Eclipse and JRRT test suites. Our technique detects 14 overly strong preconditions in Eclipse and 4 in JRRT.

5.2 Excerpt

This is a prepared text of the commencement address delivered by Steve Jobs, CEO of Apple Computer and of Pixar Animation Studios, on 12 June, 2005.

The commencement address is one of the more venerable – and respectable – traditions of American academia, especially at elite universities such as Stanford and Harvard. Because Steve Jobs died at such a relatively young age (56) this is destined to be regarded as a classic. But it faces stiff competition – as the list maintained by humanity.org testifies. Jobs's address is up against Barack Obama's lecture to Wesleyan University in 2008, Elie Wiesel's talk at DePaul University in 1997, Václav Havel's lecture on "Civilisation's Thin Veneer" at Harvard in 1995 and George Marshall's address to the same university in 1947 – to list just four. But Jobs's address has an unbearable poignancy just now, especially for those who knew him well. John Gruber, the blogger and technology commentator, saw him fairly recently and observed: "He looked old. Not old in a way that could be measured in years or even decades, but impossibly old. Not tired, but weary; not ill or unwell, but rather, somehow, ancient. But not his eyes. His eyes were young and bright, their weapons-grade intensity intact." The address also reveals something of Jobs's humanity, something that tended to get lost in the afterglow of Apple's astonishing corporate resurgence.

"I am honoured to be with you today at your commencement from one of the finest universities in the world. I never graduated from college. Truth be told, this is the closest I've ever gotten to a college graduation. Today I want to tell you three stories from my life. That's it. No big deal. Just three stories.

The first story is about connecting the dots.

I dropped out of Reed College [Portland, Oregon] after the first six months, but then stayed around as a drop-in for another 18 months or so before I really quit. So why did I drop out?

It started before I was born. My biological mother was a young, unwed college graduate student, and she decided to put me up for adoption. She felt very strongly that I should be adopted by college graduates, so everything was all set for me to be adopted at birth by a lawyer and his wife. Except that when I popped out they decided at the last minute that they really wanted a girl. So my parents, who were on a waiting list, got a call in the middle of the night asking: "We have an unexpected baby boy; do you want him?" They said: "Of course." My biological mother later found out that my mother had never graduated from college and that my father had never graduated from high school. She refused to sign the final adoption papers. She only relented a few months later when my parents promised that I would someday go to college.

This is interesting because for many years Jobs was extremely reluctant to discuss his family background in public. He was asked about it in the famous Playboy interview in 1985, for example, and refused point-blank to go into it.

And 17 years later I did go to college. But I naively chose a college that was almost as expensive as Stanford, and all of my working-class parents' savings were being spent on my college tuition. After six months I couldn't see the

value in it. I had no idea what I wanted to do with my life and no idea how college was going to help me figure it out. And here I was spending all of the money my parents had saved their entire life. So I decided to drop out and trust that it would all work out OK. It was pretty scary at the time, but looking back it was one of the best decisions I ever made. The minute I dropped out I could stop taking the required classes that didn't interest me, and begin dropping in on the ones that looked interesting.

One of the most interesting back stories of the modern computing and IT industry is how many of its pioneers were college dropouts. Bill Gates and Mark Zuckerberg, for example, both left Harvard in mid-term, and one of Gates's proudest boasts is that he was responsible for persuading Steve Ballmer (now Microsoft's CEO) to drop out of Stanford. The list continues with Larry Ellison, founder of Oracle and Michael Dell, founder of the computer firm that bears his name. And although Google's co-founders, Sergey Brin and Larry Page, did graduate, they both dropped out of their PhD programmes. This rather runs against the conventional narrative – that an expensive college education and a good degree are essential prerequisites for success.

It wasn't all romantic. I didn't have a dorm room, so I slept on the floor in friends' rooms, I returned Coke bottles for the 5¢ deposits to buy food with, and I would walk the seven miles across town every Sunday night to get one good meal a week at the Hare Krishna temple. I loved it. And much of what I stumbled into by following my curiosity and intuition turned out to be priceless later on. Let me give you one example:

Reed College at that time offered perhaps the best calligraphy instruction in the country. Throughout the campus every poster, every label on every drawer, was beautifully hand calligraphed. Because I had dropped out and didn't have to take the normal classes, I decided to take a calligraphy class to learn how to do this. I learned about serif and sans serif typefaces, about varying the amount of space between different letter combinations, about what makes great typography great. It was beautiful, historical, artistically subtle in a way that science can't capture, and I found it fascinating.

None of this had even a hope of any practical application in my life. But 10 years later, when we were designing the first Macintosh computer, it all came back to me. And we designed it all into the Mac. It was the first computer with beautiful typography. If I had never dropped in on that single course in college, the Mac would have never had multiple typefaces or proportionally spaced fonts. And since Windows just copied the Mac, it's likely that no personal computer would have them. If I had never dropped out, I would have never dropped in on this calligraphy class, and personal computers might not have the wonderful typography that they do. Of course it was impossible to connect the dots looking forward when I was in college. But it was very, very clear looking backwards 10 years later.

One of the things that everyone who knew Jobs says is that he was obsessed with "taste". He was always ranting against Microsoft because they "had no taste" and one of his great early insights was that computers would become consumer products and, as such, would have to be attractively designed and easy to use.

Again, you can't connect the dots looking forward; you can only connect them looking backwards. So you have to trust that the dots will somehow connect in your future. You have to trust in something – your gut, destiny, life, karma, whatever. This approach has never let me down, and it has made all the difference in my life.

My second story is about love and loss.

I was lucky – I found what I loved to do early in life. Woz [Steve Wozniak] and I started Apple in my parents' garage when I was 20. We worked hard, and in 10 years Apple had grown from just the two of us in a garage into a $2bn company with over 4,000 employees. We had just released our finest creation – the Macintosh – a year earlier, and I had just turned 30. And then I got fired. How can you get fired from a company you started? Well, as Apple grew we hired someone who I thought was very talented to run the company with me and for the first year or so things went well. But then our visions of the future began to diverge and eventually we had a falling-out. When we did, our board of directors sided with him. So at 30 I was out. And very publicly out. What had been the focus of my entire adult life was gone, and it was devastating.

Jobs hired John Sculley, who had been CEO of Pepsi, the drinks company, in the mistaken belief that having an experienced chief executive would free him to do what he really loved doing – which was creating great products. His pitch to Sculley to persuade him to leave Pepsi was: "John, do you really want to spend the rest of your life selling sugared water?" In the short term, appointing Sculley looked like the defining, catastrophic error of Jobs's life. But – with the 20/20 vision of hindsight – he came to see it differently. The idea that getting

fired might be a good thing is a bracing idea for young graduates, even Stanford ones.

I really didn't know what to do for a few months. I felt that I had let the previous generation of entrepreneurs down – that I had dropped the baton as it was being passed to me. I met with David Packard and Bob Noyce and tried to apologise for screwing up so badly. I was a very public failure, and I even thought about running away from the valley. But something slowly began to dawn on me – I still loved what I did. The turn of events at Apple had not changed that one bit. I had been rejected, but I was still in love. And so I decided to start over. I didn't see it then, but it turned out that getting fired from Apple was the best thing that could have ever happened to me. The heaviness of being successful was replaced by the lightness of being a beginner again, less sure about everything. It freed me to enter one of the most creative periods of my life.

During the next five years, I started a company named NeXT, another company named Pixar, and fell in love with an amazing woman who would become my wife. Pixar went on to create the world's first computer-animated feature film, Toy Story, and is now the most successful animation studio in the world. In a remarkable turn of events, Apple bought NeXT, I returned to Apple, and the technology we developed at NeXT is at the heart of Apple's current renaissance. And Laurene and I have a wonderful family together.

Again, a rare insight into his personal life. In a moving tribute his friend Stephen Wolfram recalls seeing this side of Jobs for the first time. "One time I went to see him in NeXT's swank new offices in Redwood City," he recalls. "I particularly wanted to talk to him about Mathematica as a computer language. He always preferred user interfaces to languages, but he was trying to be helpful. The conversation was going on, but he said he couldn't go to dinner, and actually he was quite distracted, because he was going out on a date that evening – and he hadn't been on a date for a long time. He explained that he'd just met the woman he was seeing a few days earlier, and was very nervous about his date. The Steve Jobs, so confident as a businessman and technologist, had melted away, and he was asking me – hardly a noted known authority on such things – about his date. As it turned out, the date apparently worked out – and within 18 months the woman he met became his wife, and remained so until the end."

I'm pretty sure none of this would have happened if I hadn't been fired from Apple. It was awful-tasting medicine, but I guess the patient needed it. Sometimes life hits you in the head with a brick. Don't lose faith. I'm convinced that the only thing that kept me going was that I loved what I did. You've got to find what you love. And that is as true for your work as it is for your lovers. Your work is going to fill a large part of your life, and the only way to be truly satisfied is to do what you believe is great work. And the only way to do great work is to love what you do. If you haven't found it yet, keep looking. Don't settle. As with all matters of the heart, you'll know when you find it. And, like any great relationship, it just gets better and better as the years roll on. So keep looking until you find it. Don't settle.

My third story is about death.

When I was 17, I read a quote that went something like: "If you live each day as if it was your last, some day you'll most certainly be right." It made an impression on me, and since then, for the past 33 years, I have looked in the mirror every morning and asked myself: "If today were the last day of my life, would I want to do what I am about to do today?" And whenever the answer has been "no" for too many days in a row, I know I need to change something.

Remembering that I'll be dead soon is the most important tool I've ever encountered to help me make the big choices in life. Because almost everything – all external expectations, all pride, all fear of embarrassment or failure – these things just fall away in the face of death, leaving only what is truly important. Remembering that you are going to die is the best way I know to avoid the trap of thinking you have something to lose. You are already naked. There is no reason not to follow your heart.

About a year ago I was diagnosed with cancer. I had a scan at 7.30 in the morning and it clearly showed a tumour on my pancreas. I didn't even know what a pancreas was. The doctors told me this was almost certainly a type of cancer that is incurable and that I should expect to live no longer than three to six months. My doctor advised me to go home and get my affairs in order, which is doctor's code for "prepare to die". It means to try to tell your kids everything you thought you'd have the next 10 years to tell them in just a few months. It means to make sure everything is buttoned up so that it will be as easy as possible for your family. It means to say your goodbyes.

I lived with that diagnosis all day. Later that evening I had a biopsy, where they stuck an endoscope down my throat, through my stomach and into my intestines, put a needle into my pancreas and got a few cells from the tumour. I was

sedated, but my wife, who was there, told me that when they viewed the cells under a microscope the doctors started crying because it turned out to be a very rare form of pancreatic cancer that is curable with surgery. I had the surgery and I'm fine now.

This was the closest I've been to facing death, and I hope it's the closest I get for a few more decades. Having lived through it, I can now say this to you with a bit more certainty than when death was a useful, but purely intellectual, concept:

No one wants to die. Even people who want to go to heaven don't want to die to get there. And yet death is the destination we all share. No one has ever escaped it. And that is as it should be, because death is very likely the single best invention of life. It is life's change agent. It clears out the old to make way for the new. Right now the new is you, but some day not too long from now, you will gradually become the old and be cleared away. Sorry to be so dramatic, but it is quite true.

This breaks the first rule of contemporary American culture – never talk about death. Work out, lose weight, don't smoke, eat carefully. It's a wonderful, liberating break from that infantile, stultifying convention. One wonders if it made his audience shift nervously in their seats.

Your time is limited, so don't waste it living someone else's life. Don't be trapped by dogma – which is living with the results of other people's thinking. Don't let the noise of others' opinions drown out your own inner voice. And, most important, have the courage to follow your heart and intuition. They somehow already know what you truly want to become. Everything else is secondary.

Jobs never allowed the opinions of others to drown out his own "inner voice". One of my favourite stories about him is about the moment when the Apple design team presented him with the first version of the iPod. He looked at it for a while, turned it over and over, weighed it in his hand and then said: "It's too big." The engineers protested that it was a miracle of state-of-the-art miniaturisation – 1,000 songs packed into that tiny space. Jobs walked over to the fish tank in the corner of his office and dropped the prototype into the water. He then pointed to the bubbles that floated from it to the surface and said: "That means there's still some space in it. It's too big." End of conversation.

When I was young, there was an amazing publication called the Whole Earth Catalog, which was one of the bibles of my generation. It was created by a fellow named Stewart Brand not far from here in Menlo Park, and he brought it to life with his poetic touch. This was in the late 1960s, before personal computers and desktop publishing, so it was all made with typewriters, scissors and Polaroid cameras. It was sort of like Google in paperback form, 35 years before Google came along: it was idealistic, and overflowing with neat tools and great notions.

Stewart and his team put out several issues of the Whole Earth Catalog, and then, when it had run its course, they put out a final issue. It was the mid-1970s, and I was your age. On the back cover of their final issue was a photograph of an early morning country road, the kind you might find yourself hitchhiking on if you were so adventurous. Beneath it were the words "Stay hungry. Stay foolish". It was their farewell message as they signed off. Stay hungry. Stay foolish. And I have always wished that for myself. And now, as you graduate to begin anew, I wish that for you. Stay hungry. Stay foolish.

Thank you all very much.

Note: This is a note. Equations within a note:

Tip: 15% if the service is good.

Example
Thing1
Thing2
Thing3

> **Warning:** Strong prose may provoke extreme mental exertion. Reader discretion is strongly advised.

You can make up your own admonition too.

```python
def some_function():
    interesting = False
    print 'This line is highlighted.'
    print 'This one is not...'
    print '...but this one is.'
```

```
As a cluster workload manager, Slurm has three key functions.
First, it allocates exclusive and/or non-exclusive access to resources (compute␣
→nodes) to users for some duration of time so they can perform work. Second, it␣
→provides a framework for starting, executing, and monitoring work (normally a␣
→parallel job) on the set of allocated nodes. Finally, it arbitrates contention for␣
→resources by managing a queue of pending work.
```

```
Some Ruby code.
Diverse perspectives brought together by a shared commitment to excellence, learning␣
→and growing Diverse perspectives brought together by a shared commitment to␣
→excellence, learning and growing Diverse perspectives brought together by a shared␣
→commitment to excellence, learning and growing
```

```
Diverse perspectives brought together by a shared commitment to excellence, learning␣
→and growing
```

5.3 Synonyms and antonyms

comprise

compose Comprise means "include", "contain" or "consist of", as in *The pie comprises eight slices*

> Be composed of means "to be or constitute a part of element of" or "to make up or form the basis of"(be made up of, in form of), as in *Eight slices compose the pie*. Besides, the use of *be composed of* more formal.
>
> The key rule to remember is that the whole comprises the elements or parts, and the elements or parts compose the whole.
>
> Examples:

```
Our wine team
comprises/includes/contains/consists of/is composed of/is constituted of/is made␣
→up of/is in form of
devoted wine lovers.
```

involve Synonyms:

```
affect, associate, catch, commit, comprise, concern, suggest
connect, contain, cover, engage, hold, link, mean, prove, require
```

Antonyms:

```
abandon, cancel, cease, divide, exclude, releae, separate, stop
ignore, reject, free, remove, explain
```

Examples:

```
1. to be the business or affiar of
This is not something that involves/affects/concerns/touches you, so do not worry␣
↪about it.
2. to have as part of a whole
A tragic play usually involves/carries/contains/embraces/entails/includes/takes␣
↪in/ a number of plot devices, including the hero's fatal flaw.
3. to hold the attention of
Her blissful daydream so completely involved her that she never heard the knock␣
↪on the door.
```

expense

cost

charge

under

below

beneath

underneath

`comprise` *comprise*

Synonyms and antonyms

Excerpt Linux
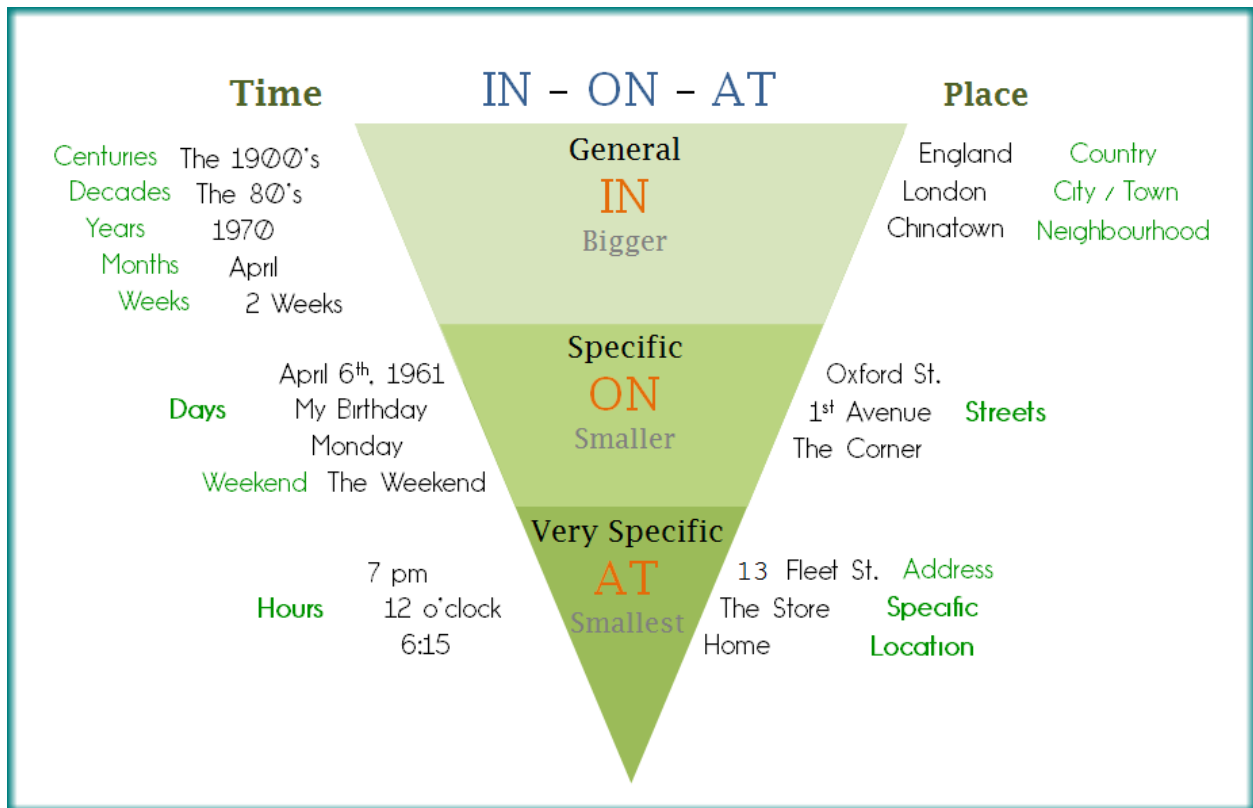
5.4 Prepositions

5.4.1 Prep. of Place

Prepositions of Place[1][2]

5.4.2 Prep. of Time

sdf[4]

---

[1] https://busyteacher.org/22990-prepositions-of-place-theory-handout.html

[2] http://www.smart-words.org/linking-words/prepositions.html

[4] http://www.smart-words.org/linking-words/prepositions.html

5.4.3 Prep. of Space

5.4.4 Prep. of in-on-at

5.5 Tenses

Simple Present	Repeated Actions	
	Facts or Generalizations	
	Scheduled Events in the Near Future	
	Now (Non-Continuous Verbs)	
Past Continuous	Interrupted Action in the Past	
	Specific Time as an Interruption	
	Parallel Actions	
	Repetition and Irritation with "Always"	
Present Perfect	Unspecified Time Before Now	
	Duration From the Past Until Now (Non-Continuous Verbs)	
Present Perfect Continuous	Duration from the Past Until Now	
	Recently, Lately	
Past Perfect	Completed Action Before Something in the Past	

[1]

---

[1] http://languagelearningbase.com/87327/english-verb-tenses-correct-tense-usage

Simple Future	"Will" to Express a Voluntary Action "Will" to Express a Promise "Be going to" to Express a Plan "Will" or "Be Going to" to Express a Prediction	
Future Continuous	Interrupted Action in the Future Specific Time as an Interruption in the Future	
	Parallel Actions in the Future	
Future Perfect	Completed Action Before Something in the Future	
	Duration Before Something in the Future (Non-Continuous Verbs)	
Future Perfect Continuous	Duration Before Something in the Future Cause of Something in the Future	

timeline	tense	use
	simple present	repeated actions
		generalizations
	present continuous	actions happening now
		actions happening around now
	simple past	completed action
		how long
	past continuous	interrupted action
	present perfect	unspecified time
		how long
	present perfect continuous	how long
	past perfect	past before the past
		how long
	past perfect continuous	how long
	future	
	future perfect	completed action before future action
	future perfect continuous	how long before future action
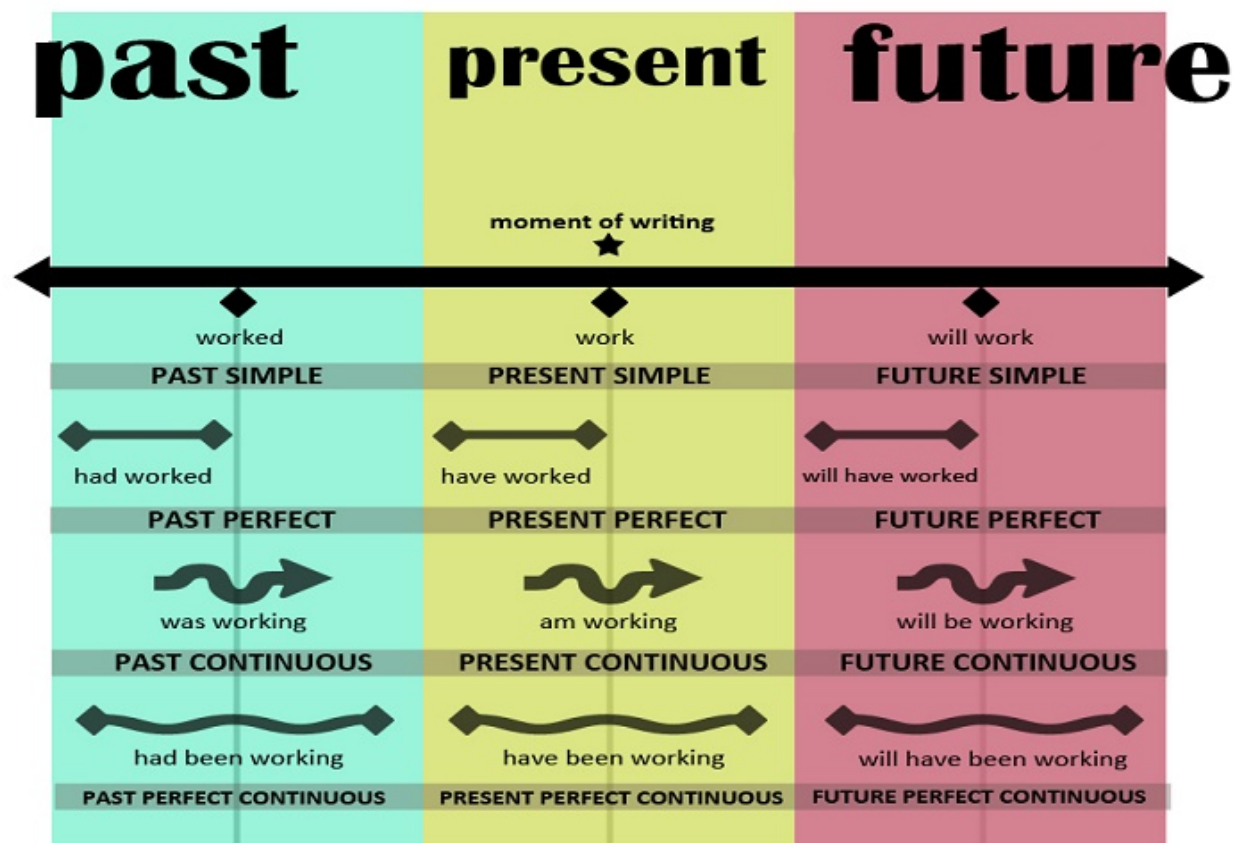
Topic Title

Subsequent indented lines comprise the body of the topic, and are interpreted as body elements.

Sidebar Title

Optional Sidebar Subtitle

Subsequent indented lines comprise the body of the sidebar, and are interpreted as body elements.

Contents

- *English*

Tip:

1. You should do your job in such a fashion that others can build on top of it, so they will indeed say, "Yes, I've stood on so and so's shoulders and I saw further."

2. When I first started, I got practically physically ill while giving a speech, and I was very, very nervous. I realized I either had to learn to give speeches smoothly or I would essentially partially **cripple** my whole career.

3. Most great scientists are completely committed to their problem. Those who don't become committed **seldom** produce outstanding, first-class work.[1]

4. And you're aware your dreams are, **to a fair extent** (), a reworking of the experiences of the day.

5. I love that place **a little bit too much** .

6. since pods may come and go as a result of failure or changes in the cardinality of the replication controller.

---

[1] http://www.cs.virginia.edu/~robins/YouAndYourResearch.html

Compilers

A Record of reStructuredText Syntax Alternatives

Quick reStructuredText

http://www.sphinx-doc.org

Index

B

below, **15**
beneath, **15**

C

charge, **15**
compose, **14**
comprise, **14**
cost, **15**

E

expense, **15**

I

involve, **14**

U

under, **15**
underneath, **15**