**This is just me compiling links from the web, here, and trying to learn stuff on my own. I suggest instead of bookmarks you make a list as well for future use. Cause there is just a lot to know.**

**My discord username is zendrulat#2829**

**@ZacharyEndrulat on Twitter if you need anything.**

|  |  |
| --- | --- |
| **Advocates**  **Links** | |
| Dave Chaney/Russ Cox/Gophercon | <https://dave.cheney.net/resources-for-new-go-programmers>   |  | | --- | | <https://research.swtch.com> | | <https://blog.gopheracademy.com> | | <https://github.com/golang/go/wiki/NewSpeakers>  Resources are at the bottom | |
|  |  |
|
| **Todd mcleod** | <https://github.com/GoesToEleven?tab=repositories>  <https://www.greatercommons.com>  <https://www.youtube.com/user/toddmcleod> |
| **William Kennedy** | <https://www.goinggo.net>  <http://www.informit.com/store/ultimate-go-programming-livelessons-9780134757483> |
| **Wiki** | <https://github.com/golang/go/wiki/Learn> |
| **Concurrency** | <https://github.com/golang/go/wiki/LearnConcurrency> |
| **Best practices** | <https://peter.bourgon.org/go-best-practices-2016/>  <https://github.com/golang-standards/project-layout/blob/master/README.md> |
| **Good resource for programming and go** | <https://www.class-central.com/course/coursera-functions-methods-and-interfaces-in-go-12050> |
| **Design patterns** | <https://github.com/tmrts/go-patterns/blob/master/README.md> |

|  |  |
| --- | --- |
| The Best Biggest Resources | |
| Best for interactive | <https://tour.golang.org/welcome/1> |
| Best for directions on what’s right | [https://golang.org/doc/effective\_go.html](https://golang.org/doc/effective_go.html#Getters) |
| Best references | [https://gowebexamples.com](https://gowebexamples.com/)  <https://github.com/GoesToEleven?tab=repositories>  <https://www.youtube.com/user/toddmcleod>  <https://astaxie.gitbooks.io/build-web-application-with-golang/content/en/03.2.html> |

|  |  |
| --- | --- |
| Beginner | |
| Example problems | <https://tour.golang.org/welcome/1> |
| Cliff notes of docs | <https://golang.org/doc/effective_go.html> |
| Go examples | [https://gowebexamples.com](https://gowebexamples.com/)  <https://learn-anything.xyz/1601>  <https://github.com/smileisak/go-training/blob/master/README.md> |
| Go by examples | <https://gobyexample.com> |
| Tutorials | <https://astaxie.gitbooks.io/build-web-application-with-golang/content/en/>  <https://hackr.io/tutorials/learn-golang>  <http://exercism.io/languages/go/about>  <http://yourbasic.org/golang/>  [https://www.getrevue.co/profile/golang/issues/autocomplete-with-go-a-github-user-search-app-sound-recording-more-133570](https://www.getrevue.co/profile/golang/issues/autocomplete-with-go-a-github-user-search-app-sound-recording-more-133570?utm_campaign=Issue&utm_content=view_in_browser&utm_medium=email&utm_source=Master+the+World+of+Golang)  <http://www.golangpatterns.info/home/>  <https://www.callicoder.com/categories/golang/> |
| Video learner | [https://www.udemy.com/go -programming-language/learn/v4/](https://www.udemy.com/go-programming-language/learn/v4/)  <http://www.informit.com/store/ultimate-go-programming-livelessons-9780134757483> |
| Blogs | <https://golangbot.com>  <https://blog.learngoprogramming.com>  <https://golangweekly.com/issues>  <https://blog.meetupfeed.com/go-talk-recordings-meetups-conferences-q4-2017/>  <https://www.ribice.ba>  <https://go101.org> |
| Contribute to open source for beginners. | <http://www.firsttimersonly.com>  <http://up-for-grabs.net/#/> |
| Good go | <https://medium.com/myntra-engineering/my-journey-with-golang-web-services-4d922a8c9897> |
| Help docs | <https://documentation.help/Golang/FAQ.html> |
| Trivia | [**https://github.com/golang/go/wiki/CodeReviewComments**](https://github.com/golang/go/wiki/CodeReviewComments)  [**https://www.reddit.com/r/golang/comments/46bd5h/ama\_we\_are\_the\_go\_contributors\_ask\_us\_anything/?utm\_source=reddit-android**](https://www.reddit.com/r/golang/comments/46bd5h/ama_we_are_the_go_contributors_ask_us_anything/?utm_source=reddit-android) |
| Coming from nodejs | [**https://github.com/miguelmota/golang-for-nodejs-developers/blob/master/README.md#files**](https://github.com/miguelmota/golang-for-nodejs-developers/blob/master/README.md#files) |
| Coming from python | [**https://golang-for-python-programmers.readthedocs.io/en/latest/**](https://golang-for-python-programmers.readthedocs.io/en/latest/) |

|  |  |
| --- | --- |
| Daily reading | |
| Blogs/tutorials/duplicates | <https://github.com/golang/go/wiki/Blogs>  <https://www.ardanlabs.com/blog/>  <https://golangweekly.com/issues>  <https://blog.gopheracademy.com>  <https://golangnews.com>  <https://forum.golangbridge.org>  <https://www.reddit.com/r/golang/>  <https://github.com/golang/go/wiki/ResearchPapers>  <https://jgoodall.me/2015/11/17/golang-resources.html>  <https://github.com/astaxie>  <https://github.com/thewhitetulip/web-dev-golang-anti-textbook/blob/master/README.md> |

|  |  |
| --- | --- |
| **Bytes** | |
| byte | <https://www.joelonsoftware.com/2003/10/08/the-absolute-minimum-every-software-developer-absolutely-positively-must-know-about-unicode-and-character-sets-no-excuses/> |
| slices | <https://blog.golang.org/slices> |
| General strings | <https://medium.com/go-walkthrough/go-walkthrough-bytes-strings-packages-499be9f4b5bd> |
| When to use []byte or string | <http://joshua.poehls.me/2014/04/go-101-string-or-byte-slice/> |
| Structs | <https://itnext.io/structures-structs-in-go-golang-9ba6d9fad61b> |
| sync.pool | <http://www.akshaydeo.com/blog/2017/12/23/How-did-I-improve-latency-by-700-percent-using-syncPool/>  <http://dominik.honnef.co/go-tip/2014-01-10/#syncpool> |
| func() as a param | <https://newfivefour.com/golang-closures-anonymous-functions.html>  My example  <https://play.golang.org/p/BSpz69Ia4DU> |

|  |  |
| --- | --- |
| **Checking values** | |
| Printing them | <https://play.golang.org/p/vEMvsxJg0mc> |
| Spew Dump | <https://github.com/davecgh/go-spew> |
| Error context | <https://github.com/pkg/errors/blob/master/README.md> |

|  |  |
| --- | --- |
| **http package** | |
| Http package go wiki | [https://golang.org/doc/articles/wiki/#tmp](https://golang.org/doc/articles/wiki/#tmp_)3 |
| Logging http | <https://blog.questionable.services/article/http-handler-error-handling-revisited/> |

|  |  |
| --- | --- |
| **General** | |
| Channels | <https://link.medium.com/WpfOPDPb4V>  <http://guzalexander.com/2013/12/06/golang-channels-tutorial.html>  <https://imil.net/blog/2018/12/31/Understanding-golang-channel-range/>  <https://medium.com/@trevor4e/learning-gos-concurrency-through-illustrations-8c4aff603b3> |
| Install | <https://www.digitalocean.com/community/tutorials/how-to-install-go-1-6-on-ubuntu-16-04>  http[://www.hostingadvice.com/how-to/install-golang-on-ubuntu/](http://www.hostingadvice.com/how-to/install-golang-on-ubuntu/)  <https://youtu.be/XCsL89YtqCs>  <https://www.youtube.com/watch?v=5qI8z_lB5Lw> |
| Doc on generics | <https://docs.google.com/document/d/1vrAy9gMpMoS3uaVphB32uVXX4pi-HnNjkMEgyAHX4N4/mobilebasic> |
| General learning | <https://sites.google.com/site/gopatterns/>  <http://yourbasic.org/golang/inheritance-object-oriented/>  <https://forum.golangbridge.org>  <http://www.golangprograms.com/go-language/slices-in-golang-programming.html>  <http://programming.guide/go/index.html>  <https://gotools.org> |
| Interfaces | <https://medium.com/golangspec/interfaces-in-go-part-iii-61f5e7c52fb5>  <https://research.swtch.com/interfaces>  <https://www.golang-book.com/books/intro/9>  <https://www.youtube.com/watch?v=83J256zByHA>  <https://github.com/golang/go/wiki/CodeReviewComments#interfaces>  <https://play.golang.org/p/pv7o3G32I1>  <https://itnext.io/interfaces-in-golang-f1aea1ba31cb> |
| Database | <https://www.vividcortex.com/resources/the-ultimate-guide-to-building-database-driven-apps-with-go>  <http://www.alexedwards.net/blog/organising-database-access>  Tutorials - <http://go-database-sql.org/index.html>  <https://github.com/golang/go/wiki/SQLInterface>  Structs to tables - <https://github.com/samonzeweb/godb/blob/master/README.md> |
| Restful api | <http://thenewstack.io/make-a-restful-json-api-go/>  <https://tutorialedge.net/creating-simple-restful-json-api-with-go>  <https://stevenwhite.com/building-a-rest-service-with-golang-2/> - much better resource  <http://blog.narenarya.in/build-rest-api-go-mysql.html>  <https://hackernoon.com/build-restful-api-in-go-and-mongodb-5e7f2ec4be94>  <http://blog.questionable.services/article/custom-handlers-avoiding-globals/>  <https://medium.com/statuscode/how-i-write-go-http-services-after-seven-years-37c208122831>  <https://dev.to/geosoft1/versioning-your-api-in-go-1g4h?utm_campaign=Golang%20Ninjas%20Newsletter&utm_medium=email&utm_source=Revue%20newsletter> |
| GRPC  A way to mix languages. | You can skip all these by watching this.  <https://www.safaribooksonline.com/videos/complete-introduction-to/9781789349344>  Great video for advanced  <https://www.udemy.com/grpc-golang/>  <https://github.com/grpc-ecosystem/awesome-grpc/blob/master/readme.md>  My kinda learning GRPC writeup because it was taking up a whole page on here.  <https://docs.google.com/document/d/1jrJHsCzokZCBPhutxjN0h7cxaYDVLSoQVS1G4V8Eggg/edit?usp=sharing> |
| Libs | <https://github.com/avelino/awesome-go> |
| Go gotchas | <http://devs.cloudimmunity.com/gotchas-and-common-mistakes-in-go-golang/>  <https://twitter.com/davecheney/status/604837853344989184%3Flang%3Den&ved=2ahUKEwiW6tCVmvvYAhWG6oMKHYzJBlMQjjgwBHoECAUQAQ&usg=AOvVaw0zNs3pPbfn6pcKEQ2vPLbV> |
| Organization | <https://peter.bourgon.org/go-best-practices-2016/#repository-structure>  <https://github.com/golang-standards/project-layout>  <https://medium.com/@benbjohnson/standard-package-layout-7cdbc8391fc1> |
| Imports | <https://medium.com/golangspec/import-declarations-in-go-8de0fd3ae8ff> |
| Slices/Data structures | <https://blog.golang.org/slices>  <https://blog.golang.org/go-slices-usage-and-internals>  <https://research.swtch.com/godata>  <https://www.ardanlabs.com/blog/2013/08/understanding-slices-in-go-programming.html>  <https://nanxiao.gitbooks.io/golang-101-hacks/content/posts/the-internals-of-slice.html> |
| Go GUIs | <https://github.com/tomarus/go-gui-libs/wiki> |
| Folder structure | <https://rakyll.org/style-packages/>  <https://medium.com/@benbjohnson/standard-package-layout-7cdbc8391fc1#.ds38va3pp>  <https://peter.bourgon.org/go-best-practices-2016/#repository-structure>  <https://www.goinggo.net/2017/02/design-philosophy-on-packaging.html>  [https://www.elliotdwright.com/2018/02/27/directory-structure-and-manual-wiring/amp/](https://www.elliotdwright.com/2018/02/27/directory-structure-and-manual-wiring/amp/?__twitter_impression=true)  <https://talks.bjk.fyi/bketelsen/talks?p=gcru18-best#/1> |
| Concurrency | <https://medium.com/@matryer/very-basic-concurrency-for-beginners-in-go-663e63c6ba07> |
| context | <https://www.jtolio.com/2017/01/writing-advanced-web-applications-with-go/> |
| Docker | <https://youtu.be/UtNeLfXaBJM> |
| Code review | <https://github.com/golang/go/wiki/CodeReviewComments> |
| Auth2 | <https://hackernoon.com/build-your-own-oauth2-server-in-go-7d0f660732c3> |

|  |  |
| --- | --- |
| Beginner building | |
| Middleware/logcolors/context | <https://www.reddit.com/r/golang/comments/7p35s4/how_do_i_get_the_response_status_for_my_middleware/>  <https://github.com/logrusorgru/aurora>  <https://gocodecloud.com/blog/2016/11/15/simple-golang-http-request-context-example/>  <https://gitlab.com/zendrulat123/prettymiddleware/wikis/home> |
| Really good form data from golang | <https://dinosaurscode.xyz/go/2016/06/19/golang-mysql-authentication/> |
| Templates and routes | <https://github.com/GoesToEleven/golang-web-dev/blob/master/021_third-party-serveMux/01_julienschimdt/main.go>  <https://blog.gopheracademy.com/advent-2017/using-go-templates/> |
| Connecting Jqeury/other asset files with Mux router | <https://stackoverflow.com/questions/15834278/serving-static-content-with-a-root-url-with-the-gorilla-toolkit> |
| CORS (preflight error with jqeury) | <https://stackoverflow.com/questions/39507065/enable-cors-in-golang> |
| Ajax requests with jqeury | <https://blog.garstasio.com/you-dont-need-jquery/ajax/> |
| Design | <https://dave.cheney.net/2016/08/20/solid-go-design> |
| Go books | <https://github.com/golang/go/wiki/Books>  <https://www.programming-books.io/essential/go/>  <https://github.com/dgryski/go-perfbook/blob/master/README.md>  <https://astaxie.gitbooks.io/build-web-application-with-golang/en/preface.html> |
| cookies | <https://www.calhoun.io/securing-cookies-in-go/>  <https://austingwalters.com/building-a-web-server-in-go-web-cookies/> |
| Curl commands to go | <https://mholt.github.io/curl-to-go/> |
| Docker | <https://link.medium.com/X1xQhMFnfU> |

|  |  |
| --- | --- |
| Framework Examples | |
| GIn | <https://semaphoreci.com/community/tutorials/building-go-web-applications-and-microservices-using-gin> |
| Mux | <https://github.com/somehuman22/userapi> |

|  |  |
| --- | --- |
| Testing Go code | |
| General testing | <https://quii.gitbook.io/learn-go-with-tests/go-fundamentals/hello-world>  <https://medium.com/@matryer/5-simple-tips-and-tricks-for-writing-unit-tests-in-golang-619653f90742>  https[://blog.alexellis.io/golang-writing-unit-tests/](https://blog.alexellis.io/golang-writing-unit-tests/)  <https://medium.com/@povilasve/go-advanced-tips-tricks-a872503ac859>  <https://dev.to/quii/learn-go-by-writing-tests--m63>  <https://medium.com/@pierreprinetti/a-pattern-for-go-tests-3468b51535> |
| Interface Testing | <https://syslog.ravelin.com/what-is-an-interface-ee67b3cc9e97> |
| Profiling | <https://jvns.ca/blog/2017/09/24/profiling-go-with-pprof/> |

|  |  |
| --- | --- |
| Reader Type | |
| io.Reader | <https://medium.com/@matryer/golang-advent-calendar-day-seventeen-io-reader-in-depth-6f744bb4320b> |
| example | <https://tour.golang.org/methods/21> |

|  |  |
| --- | --- |
| Context/logging | |
| context | <https://stackoverflow.com/questions/40379960/golang-context-withvalue-how-to-add-several-key-value-pairs>  <https://play.golang.org/p/JtDAj7aVIH> |
| example | <https://play.golang.org/p/xrK0j2W3Ecs> |

|  |  |
| --- | --- |
| Interfaces 2 | |
| Russ’s explanation | <https://research.swtch.com/interfaces> |
| Understanding Interfaces | <http://jordanorelli.com/post/32665860244/how-to-use-interfaces-in-go> |
| Basic usage | <https://codeburst.io/understanding-solid-principles-interface-segregation-principle-b2d57026cf6c> |
| Best practices | <https://blog.learngoprogramming.com/golang-funcs-params-named-result-values-types-pass-by-value-67f4374d9c0a> |
| Open interface | <http://changelog.ca/log/2015/01/30/golang> |
| Using open interface for api | <https://blog.gopheracademy.com/advent-2016/advanced-encoding-decoding/>  <http://polyglot.ninja/golang-json/> |

|  |  |
| --- | --- |
| Why’s | |
| Why inheritance questions | <https://golang.org/doc/faq#inheritance> |

|  |  |
| --- | --- |
| Others lists | |
| Compilation | <https://github.com/enocom/gopher-reading-list/blob/master/README.md> |
|  |  |

|  |  |
| --- | --- |
| Syntax | |
| Am8 cheat sheet | <https://github.com/a8m/go-lang-cheat-sheet> |
| Conversions | <http://yourbasic.org/golang/conversions/> |
| []string to string | https://golang.org/pkg/strings/#Join |
| concatenation | <https://www.calhoun.io/6-tips-for-using-strings-in-go/> |

|  |  |
| --- | --- |
| Go Wasm | |
| links | <https://github.com/golang/go/wiki/WebAssembly> |

|  |  |
| --- | --- |
| My Own | |
| My Site | [ttp://159.89.238.110:8080/resources](http://159.89.238.110:8080/resources)  Sometimes I forget to leave it running |
| Slices/Arrays - with structs | <https://play.golang.org/p/DhT__tgtd6> |
| Interface magic | <https://play.golang.org/p/IA09xBX7dYF> |
| Embedding magic | <https://play.golang.org/p/7UZUogIW_n-> |
| **The Reader Interface for Go** | <https://medium.com/@snippet22/reader-types-2ba61290a5cf> |
| **Go Errors** | <https://medium.com/@snippet22/errors-in-go-1ebfa1c1b883> |
| **Messing with go wasm slowly** | <https://play.golang.org/p/d_UziITZdoj> |
| **Restful simple example** | <https://gitlab.com/zendrulat123/goes> |
| **Start of my server logger** | <https://gitlab.com/zendrulat123/prettylogserver/blob/master/showServer.go> |
| **Pretty Middleware Logger** | <https://gitlab.com/zendrulat123/prettymiddleware/wikis/home> |

|  |  |
| --- | --- |
| Example projects | |
| 1. | <https://github.com/josephspurrier/gowebapp> |

|  |  |
| --- | --- |
| Resources on scaffolding restful api from url | |
| Sending form data | <https://developer.mozilla.org/en-US/docs/Learn/HTML/Forms/Sending_and_retrieving_form_data> |
| Use curl commands first instead of building form or query. | curl -i 'http://youripaddress/get?link=www.google.com&meta=search' -H 'Content-Type: text/pl&meta=searchin' -X POST -d "some post data" |
| Restful resource | <https://www.programming-books.io/essential/go/a-21219-handling-http-method-accessing-query-strings-request-body> |
| Sending request jqeury/ajax without refresh | <https://code.tutsplus.com/tutorials/submit-a-form-without-page-refresh-using-jquery--net-59> |
| Things to remember | for preflight  1. check with https://jqueryui.com/autocomplete/ to see if the jqeury is working  2. make sure the jqeury url: "http:url" is complete.  3. make sure you are using all the properties of the struct and you are NOT decoding just the properties  4. make sure to set the route to cors.  handler := http.HandlerFunc(links)  c := cors.New(cors.Options{  AllowedOrigins: []string{"\*"},  })  handler = http.HandlerFunc(show) |
| Jqeury json parsing | https://www.digitalocean.com/community/tutorials/how-to-work-with-json-in-javascript |
| Jqeury autocomple | https://jqueryui.com/autocomplete/ |

|  |  |
| --- | --- |
| Mocking XML | |
| Mocking xml | To create mock  <http://www.json-gen.com/>  <https://onlinerandomtools.com/generate-random-xml?&depth=1&max-elements=10&always-max-elements=true>  To understand consuming xml  <https://www.programming-books.io/essential/go/6744c6d0d620448dbe66e224f64b6f8b-xml>  To consume a json resource in go  <https://tutorialedge.net/golang/consuming-restful-api-with-go/>  To consume a xml response in go  <https://larry-price.com/blog/2015/12/04/xml-parsing-in-go/>  To create structs from mock  <https://www.onlinetool.io/xmltogo/>  An example put together  <https://play.golang.org/p/hvlcsPAbUmL> |

|  |  |
| --- | --- |
| Non Go links | |
| Front-end | <https://github.com/h5bp/Front-end-Developer-Interview-Questions>  <https://github.com/yangshun/front-end-interview-handbook>  <https://frontendmasters.com/books/front-end-handbook/2018/2018.html>  <https://github.com/getify/You-Dont-Know-JS/blob/master/README.md>  <https://wesbos.com/>  <https://www.youtube.com/user/TechGuyWeb>  <https://frontendmasters.com/books/front-end-handbook/2019/#3> |
| Position | <https://css-tricks.com/absolute-relative-fixed-positioining-how-do-they-differ/> |
| Margins | <https://www.sitepoint.com/collapsing-margins/> |
| Javascript quick playground for testing quickly | <https://stephengrider.github.io/JSPlaygrounds/> |
| Removing characters with javascript for api | <https://jsfiddle.net/7tuL3cac/1/> |
| Design tips | <https://medium.com/daily-now/most-popular-development-posts-of-the-week-11-apr-2018-12f7f2e1c4a>  <https://www.youtube.com/playlist?list=PLmZLFWEwl1DQBMjKSI5sqFuGCxZSJC3f4> |
| Web assembly | <https://hacks.mozilla.org/2017/02/creating-and-working-with-webassembly-modules/>  <https://github.com/golang/go/wiki/WebAssembly> |
| Git reference | <https://excalith.github.io/Interactive-Git-Cheatsheet/>  <https://learngitbranching.js.org> |
| Web resources | <https://www.reddit.com/r/FreelanceProgramming/comments/8r3a1p/comprehensive_list_of_all_kinds_of_resources_for/?utm_source=reddit-android>  <https://github.com/leonardomso/33-js-concepts/blob/master/README.md>  Top 300 mentioned ​books on stackoverflow.com by 📚Bookadvice <https://link.medium.com/OIwMUPilbR> |
| Js | <https://flaviocopes.com/javascript/>  <https://javascript.info>  <http://www.javascriptkit.com> |
| Responsive squares just css | <https://codepen.io/anon/pen/djKBgq> |
| gitlab | cloud9  <https://gist.github.com/juemura/899241d73cf719de7f540fc68071bd7d>  <https://learngitbranching.js.org> |
| Turn any web page into an epub file. | <https://play.google.com/store/apps/details?id=com.softesk.grabMyBooksFull>  <https://www.grabmybooks.com> |
| Math | <http://www.math.ucla.edu/~tao/QED/QED.html> |
| Acme editor | <https://github.com/evbogdanov/acme>  Go version of acme  <https://github.com/as/a> |
| General coding | <https://github.com/DovAmir/awesome-design-patterns/blob/master/README.md>  <https://github.com/sindresorhus/awesome/blob/master/readme.md>  <https://www.reddit.com/r/learnprogramming/comments/2a9ygh/1000_beginner_programming_projects_xpost/> |
| Computer science | <https://github.com/ossu/computer-science/blob/dev/README.md>  <https://www.extension.harvard.edu/open-learning-initiative/intensive-introduction-computer-science> |
| Cheat sheets | <https://link.medium.com/8lXSDySLTT>  <https://www.codeinwp.com/blog/wordpress-cheat-sheets-web-development-design/?amp>  <https://envato.com/blog/cheatsheets-web-designers/>  <http://overapi.com>  <https://gto76.github.io/python-cheatsheet/>  <https://devhints.io/> |
| CSP's | <https://en.m.wikipedia.org/wiki/Constraint_satisfaction_problem>  <https://github.com/gnboorse/centipede>  <https://en.m.wikipedia.org/wiki/Quicksort> |
| Algorithms | <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/lecture-videos/> |
| Coding problems | <https://codingcompetitions.withgoogle.com/codejam/archive> |

|  |  |
| --- | --- |
| VUEJS | |
| Basic tutorial | The guy who made these videos apparently likes Go  <https://scrimba.com/g/glearnvue>  <https://learning.oreilly.com/search/?query=vuejs> |
| Learning example todo list | <http://jsfiddle.net/zendrulat/eywraw8t/525401/> |