Google techs for developers

JIEUNG KIM

jieungkim@inha.ac.kr

인하대학교
INHA UNIVERSITY

Contents

- Who am I?
- Websites and Youtube channels for Google techs and lives
- Google technologies: Programming language style guides
- Google technologies: Googletest framework
- Google technologies: Machine learning related tools
    - Colab
    - Tensorflow and Keras
    - Tensorflow model optimization toolkit
    - Tensorflow hub and model garden
    - Tensorboard
- Summary

Who am I?

인하대학교
INHA UNIVERSITY

Who am I?

- **Who am I? – https://jieung.kim**
  - I am an **assistant professor** in **Computer Engineering Department**, College of Software and Convergence, Inha University (Incheon, South Korea)
  - Before that, I was a **research engineer** at Google
    - Worked on **machine learning model optimizations** (2022.03~2022.08)
    - Worked on **pKVM formal verification** (2020.05~2022.02)
      - **pKVM**: a new software stack in the Android ecosystem to increase security
      - **Formal verification**: the strongest method to show the correctness and security properties of software with using mathematical and computational logic methods
  - Even before that, I was at Yale University as a **graduate student**, worked on other **formal verification projects**
    - Verify OS & hypervisor based on xv6 (CertiKOS)
    - Provide unified and verified APIs for distributed protocol (ADO)
  - I received my M.S. and B.S. from KAIST and SKKU (South Korea), respectively

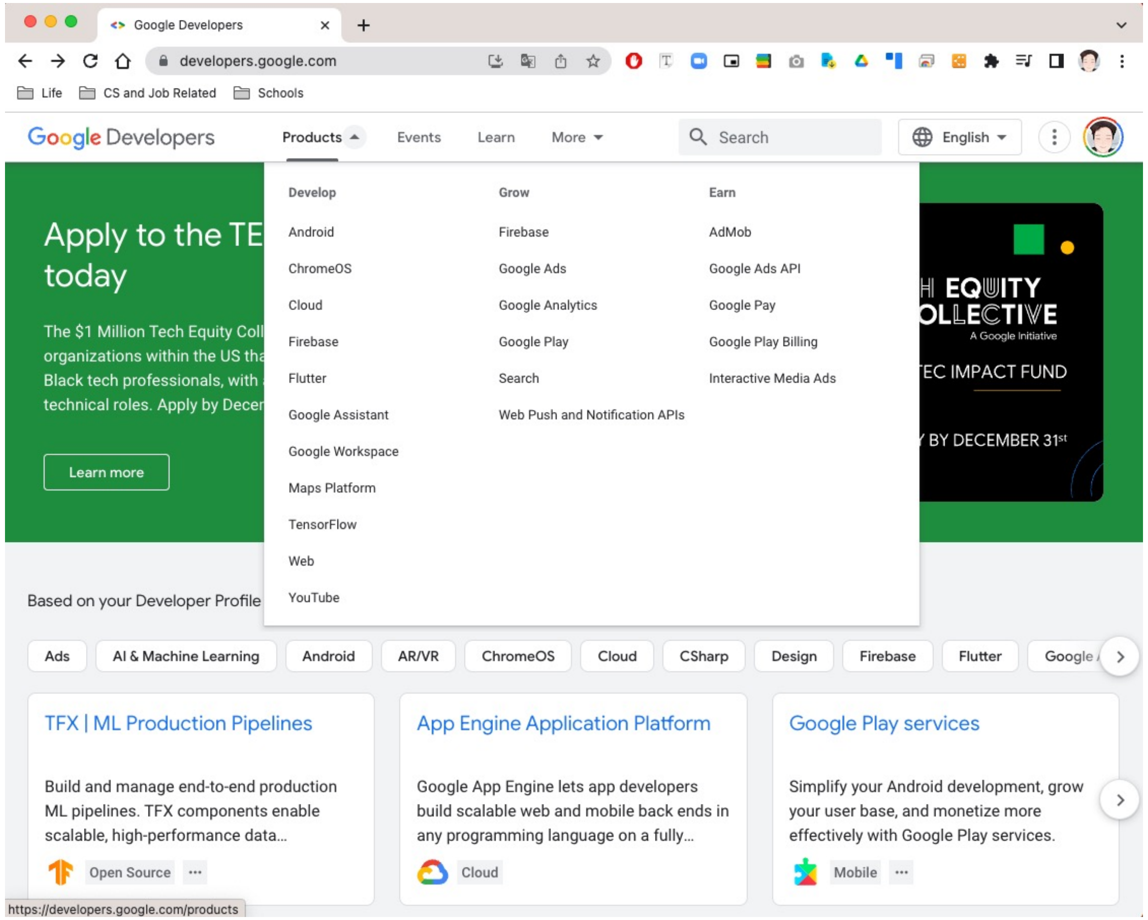
인하대학교
INHA UNIVERSITY

Websites and Youtube channels for Google techs and lives

인하대학교
INHA UNIVERSITY

Google developers

https://developers.google.com/



Apply to the TE
today

The $1 Million Tech Equity Coll
organizations within the US tha
Black tech professionals, with
technical roles. Apply by Decer

Based on your Developer Profile

TFX | ML Production Pipelines

Build and manage end-to-end production ML pipelines. TFX components enable scalable, high-performance data...

App Engine Application Platform

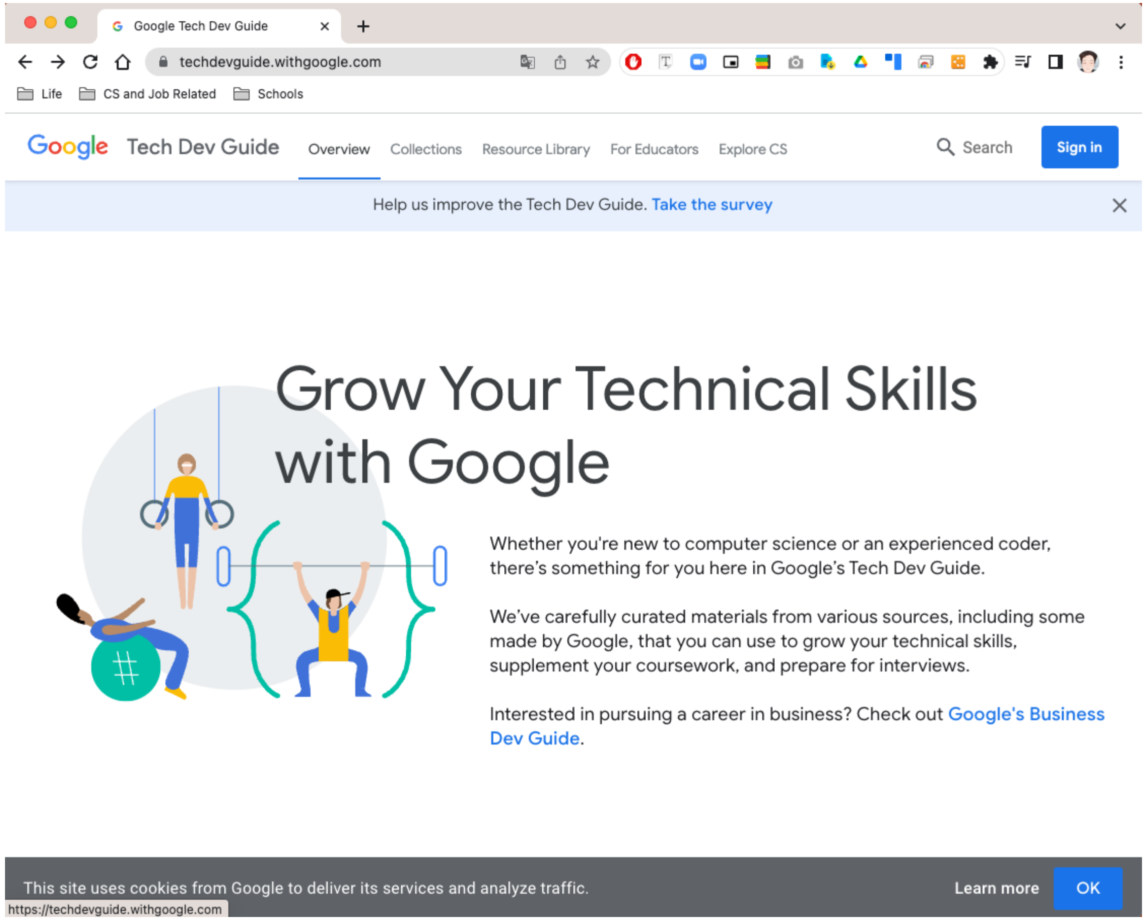
Google App Engine lets app developers build scalable web and mobile back ends in any programming language on a fully...

Google Play services

Simplify your Android development, grow your user base, and monetize more effectively with Google Play services.

Tech dev guide

https://techdevguide.withgoogle.com/



Help us improve the Tech Dev Guide. Take the survey

Grow Your Technical Skills with Google

Whether you're new to computer science or an experienced coder, there's something for you here in Google's Tech Dev Guide.

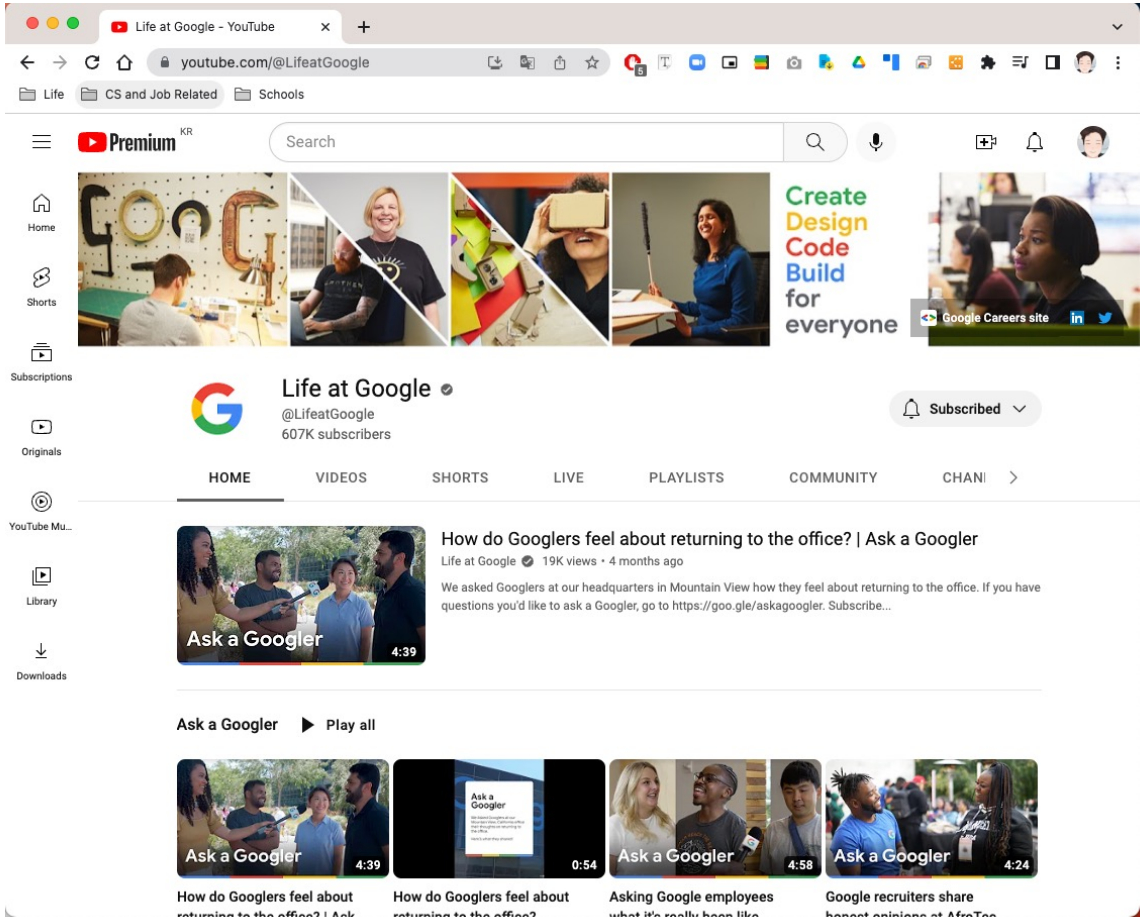
We've carefully curated materials from various sources, including some made by Google, that you can use to grow your technical skills, supplement your coursework, and prepare for interviews.

Interested in pursuing a career in business? Check out Google's Business Dev Guide.

This site uses cookies from Google to deliver its services and analyze traffic.

Youtube, Life at Google

https://www.youtube.com/lifeatgoogle



Life at Google

@LifeatGoogle

607K subscribers

How do Googlers feel about returning to the office? | Ask a Googler

Life at Google · 19K views • 4 months ago

We asked Googlers at our headquarters in Mountain View how they feel about returning to the office. If you have questions you'd like to ask a Googler, go to https://goo.gle/askagoogler. Subscribe...

Ask a Googler

How do Googlers feel about

How do Googlers feel about

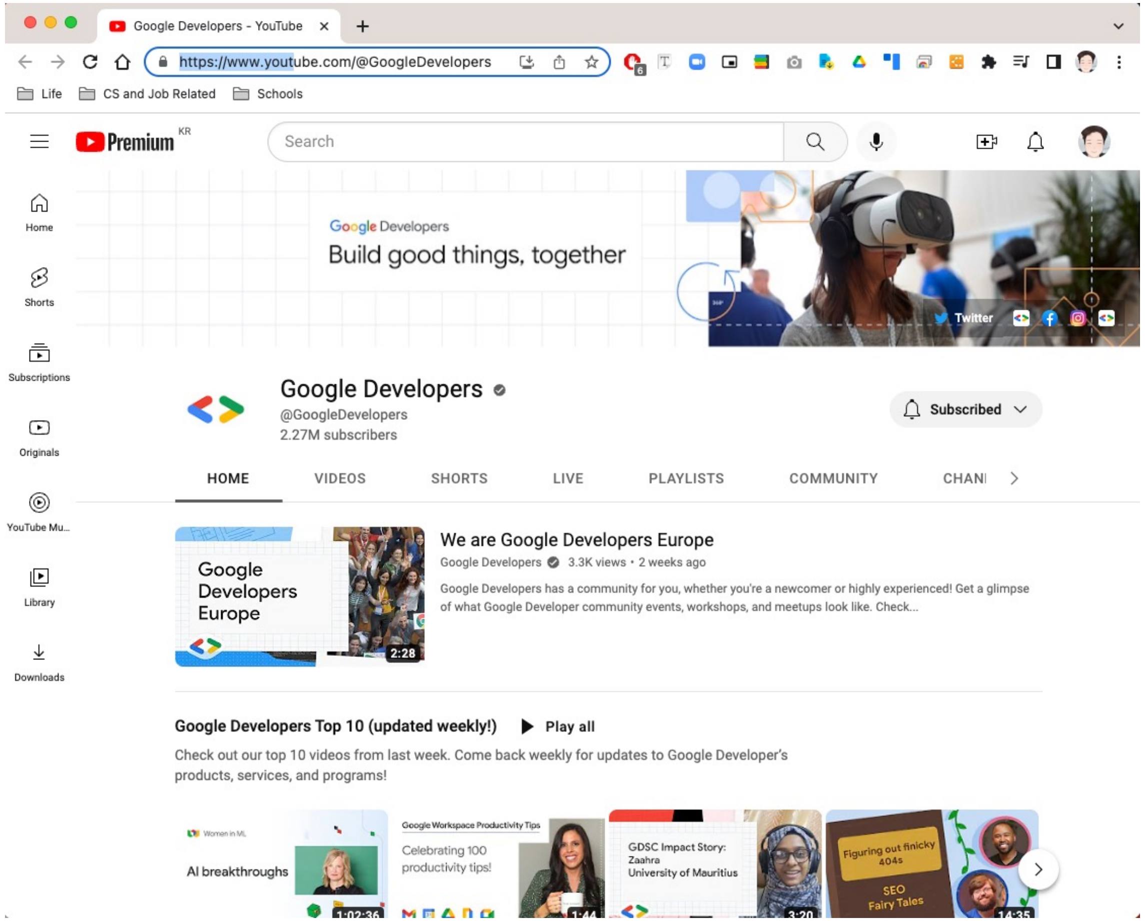
Asking Google employees

Google recruiters share

인하대학교
INHA UNIVERSITY

Youtube, Google Developers

https://www.youtube.com/GoogleDevelopers



Google Developers
@GoogleDevelopers
2.27M subscribers

We are Google Developers Europe

Google Developers · 3.3K views · 2 weeks ago

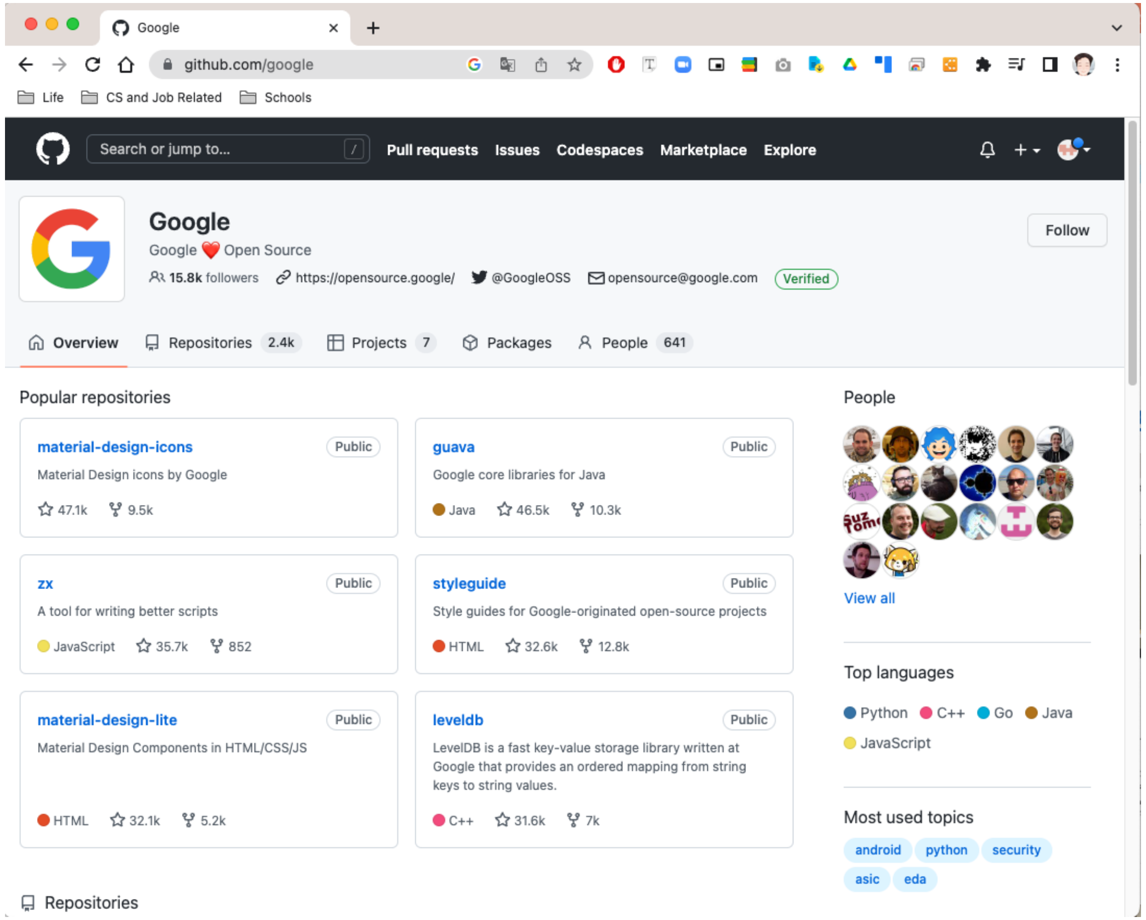
Google Developers has a community for you, whether you're a newcomer or highly experienced! Get a glimpse of what Google Developer community events, workshops, and meetups look like. Check...

Google Developers Top 10 (updated weekly!)

Check out our top 10 videos from last week. Come back weekly for updates to Google Developer's products, services, and programs!

Google repository at Github

https://github.com/google



Google

Google ❤ Open Source

15.8k followers · https://opensource.google/ · @GoogleOSS · opensource@google.com · Verified

Popular repositories

Repository	Visibility	Description	Language	Stars	Forks
material-design-icons	Public	Material Design icons by Google		47.1k	9.5k
guava	Public	Google core libraries for Java	Java	46.5k	10.3k
zx	Public	A tool for writing better scripts	JavaScript	35.7k	852
styleguide	Public	Style guides for Google-originated open-source projects	HTML	32.6k	12.8k
material-design-lite	Public	Material Design Components in HTML/CSS/JS	HTML	32.1k	5.2k
leveldb	Public	LevelDB is a fast key-value storage library written at Google that provides an ordered mapping from string keys to string values.	C++	31.6k	7k

People

View all

Top languages

Python · C++ · Go · Java · JavaScript

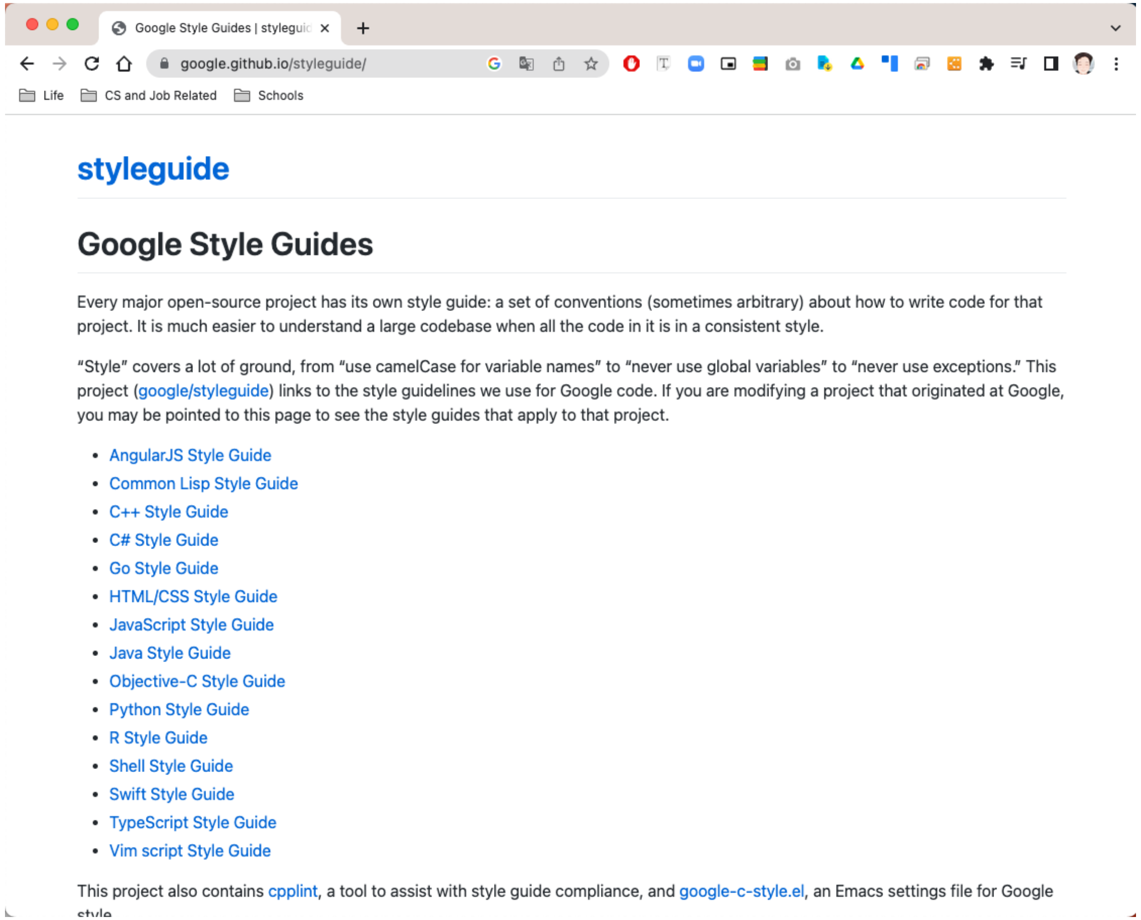
Most used topics

android · python · security · asic · eda

Repositories

Google style guide

https://google.github.io/styleguide/



styleguide

Google Style Guides

Every major open-source project has its own style guide: a set of conventions (sometimes arbitrary) about how to write code for that project. It is much easier to understand a large codebase when all the code in it is in a consistent style.

"Style" covers a lot of ground, from "use camelCase for variable names" to "never use global variables" to "never use exceptions." This project (google/styleguide) links to the style guidelines we use for Google code. If you are modifying a project that originated at Google, you may be pointed to this page to see the style guides that apply to that project.

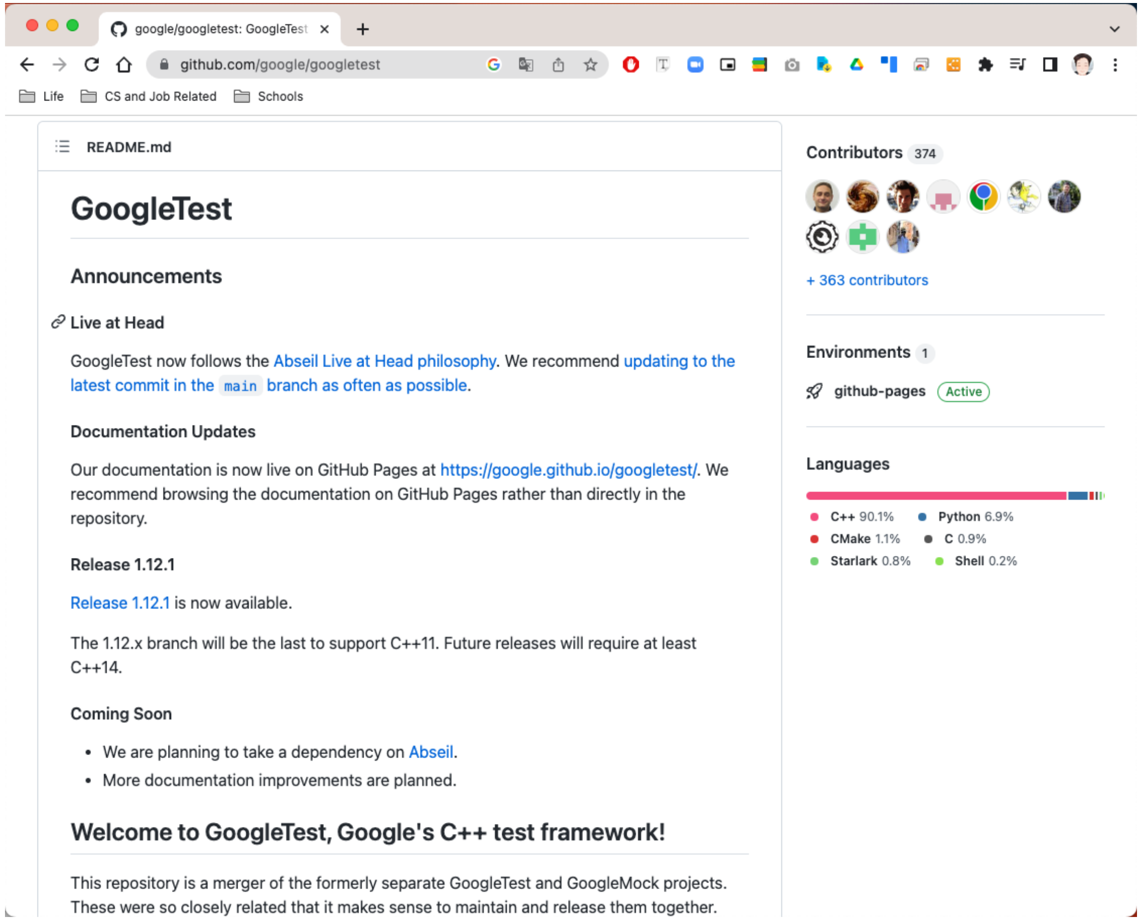
- AngularJS Style Guide
- Common Lisp Style Guide
- C++ Style Guide
- C# Style Guide
- Go Style Guide
- HTML/CSS Style Guide
- JavaScript Style Guide
- Java Style Guide
- Objective-C Style Guide
- Python Style Guide
- R Style Guide
- Shell Style Guide
- Swift Style Guide
- TypeScript Style Guide
- Vim script Style Guide

This project also contains cpplint, a tool to assist with style guide compliance, and google-c-style.el, an Emacs settings file for Google style

인하대학교
INHA UNIVERSITY

Googletest framework

https://github.com/google/googletest



GoogleTest

Announcements

Live at Head

GoogleTest now follows the Abseil Live at Head philosophy. We recommend updating to the latest commit in the `main` branch as often as possible.

Documentation Updates

Our documentation is now live on GitHub Pages at https://google.github.io/googletest/. We recommend browsing the documentation on GitHub Pages rather than directly in the repository.

Release 1.12.1

Release 1.12.1 is now available.

The 1.12.x branch will be the last to support C++11. Future releases will require at least C++14.

Coming Soon

- We are planning to take a dependency on Abseil.
- More documentation improvements are planned.

Welcome to GoogleTest, Google's C++ test framework!

This repository is a merger of the formerly separate GoogleTest and GoogleMock projects. These were so closely related that it makes sense to maintain and release them together.

Contributors 374

+ 363 contributors

Environments 1

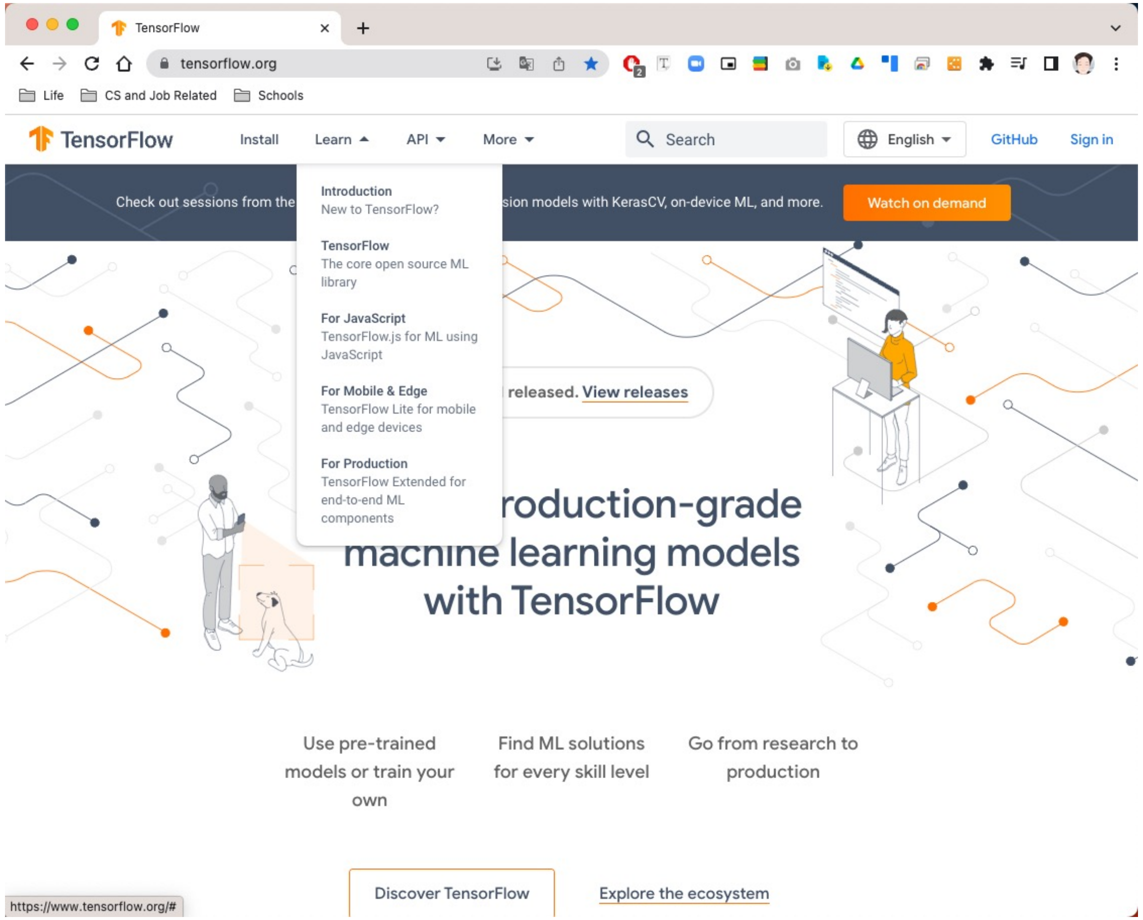
github-pages Active

Languages

● C++ 90.1%	● Python 6.9%
● CMake 1.1%	● C 0.9%
● Starlark 0.8%	● Shell 0.2%

Tensorflow

https://www.tensorflow.org/



TensorFlow

tensorflow.org

Life CS and Job Related Schools

TensorFlow Install Learn API More Search English GitHub Sign in

Check out sessions from the … sion models with KerasCV, on-device ML, and more. Watch on demand

Introduction
New to TensorFlow?

TensorFlow
The core open source ML library

For JavaScript
TensorFlow.js for ML using JavaScript

For Mobile & Edge
TensorFlow Lite for mobile and edge devices

For Production
TensorFlow Extended for end-to-end ML components

released. View releases

roduction-grade machine learning models with TensorFlow

Use pre-trained models or train your own

Find ML solutions for every skill level

Go from research to production

Discover TensorFlow

Explore the ecosystem

https://www.tensorflow.org/#

인하대학교 INHA UNIVERSITY

Google technologies: Programming language style guides

인하대학교
INHA UNIVERSITY

Programming language style guides

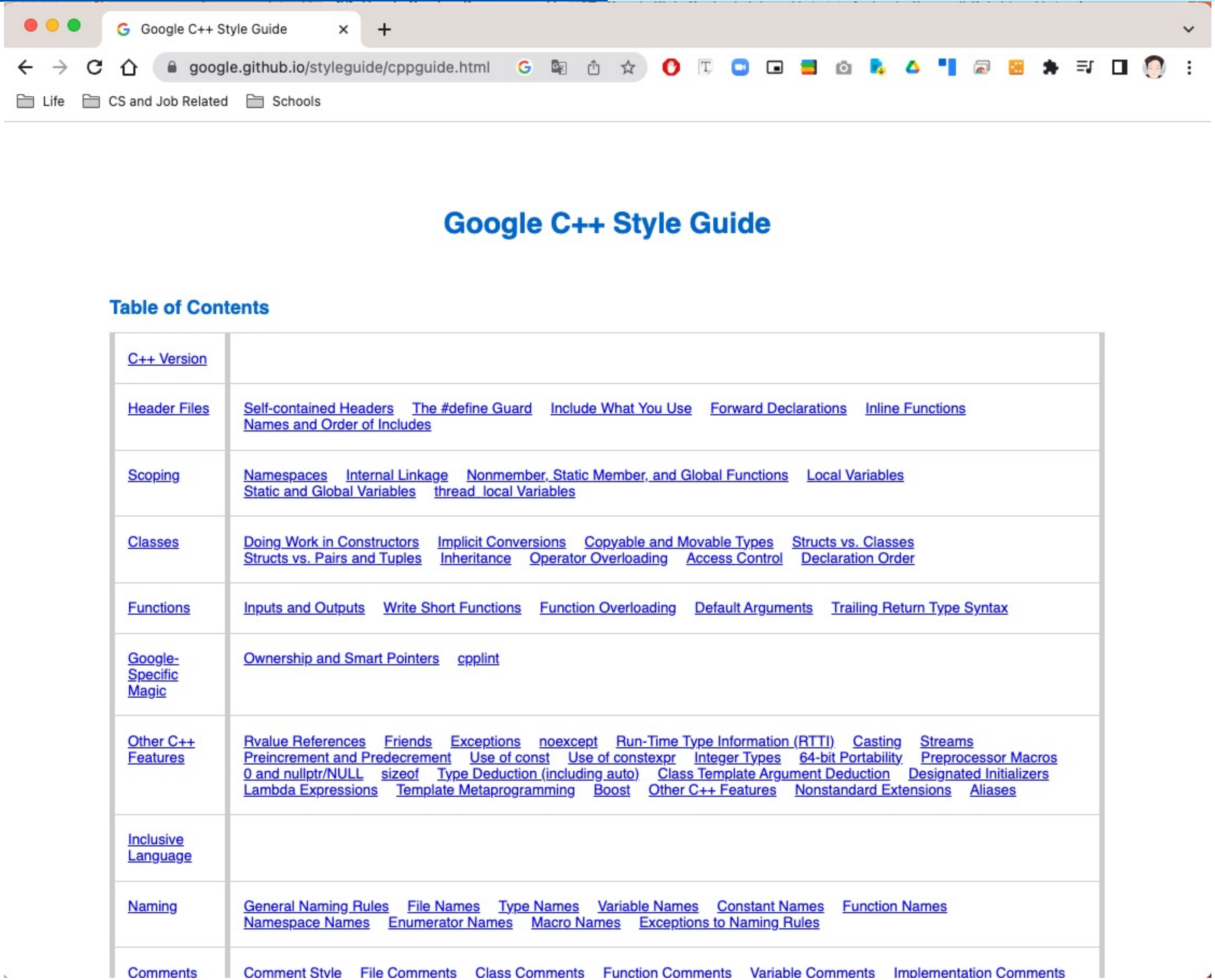
- **Programming language style guides**
  - A guide of programming conventions, style, and best practices for a team or project
  - Following the guides in the development make team's code more consistent and readable
    - Consistent code is easier to read and understand making it faster to add new features
    - **Code review** process is usually check whether the code follows style guides

Code review

- **Code review**
  - Careful, systematic study of source code by people who are not the original author of the code
- **Purpose of code review**
  - Can catch many bugs, design flaws early
  - > 1 person has seen every piece of code
    - Insurance against author’s disappearance
  - Forcing function for documentation and code improvements
    - Authors to articulate their decisions
    - Authors participate in the discovery of flaws
    - Prospect of someone reviewing your code raises quality threshold
  - Inexperienced personnel get hands-on experience without hurting code quality
    - Pairing them up with experienced developers
    - Can learn by being a reviewer as well

Code review

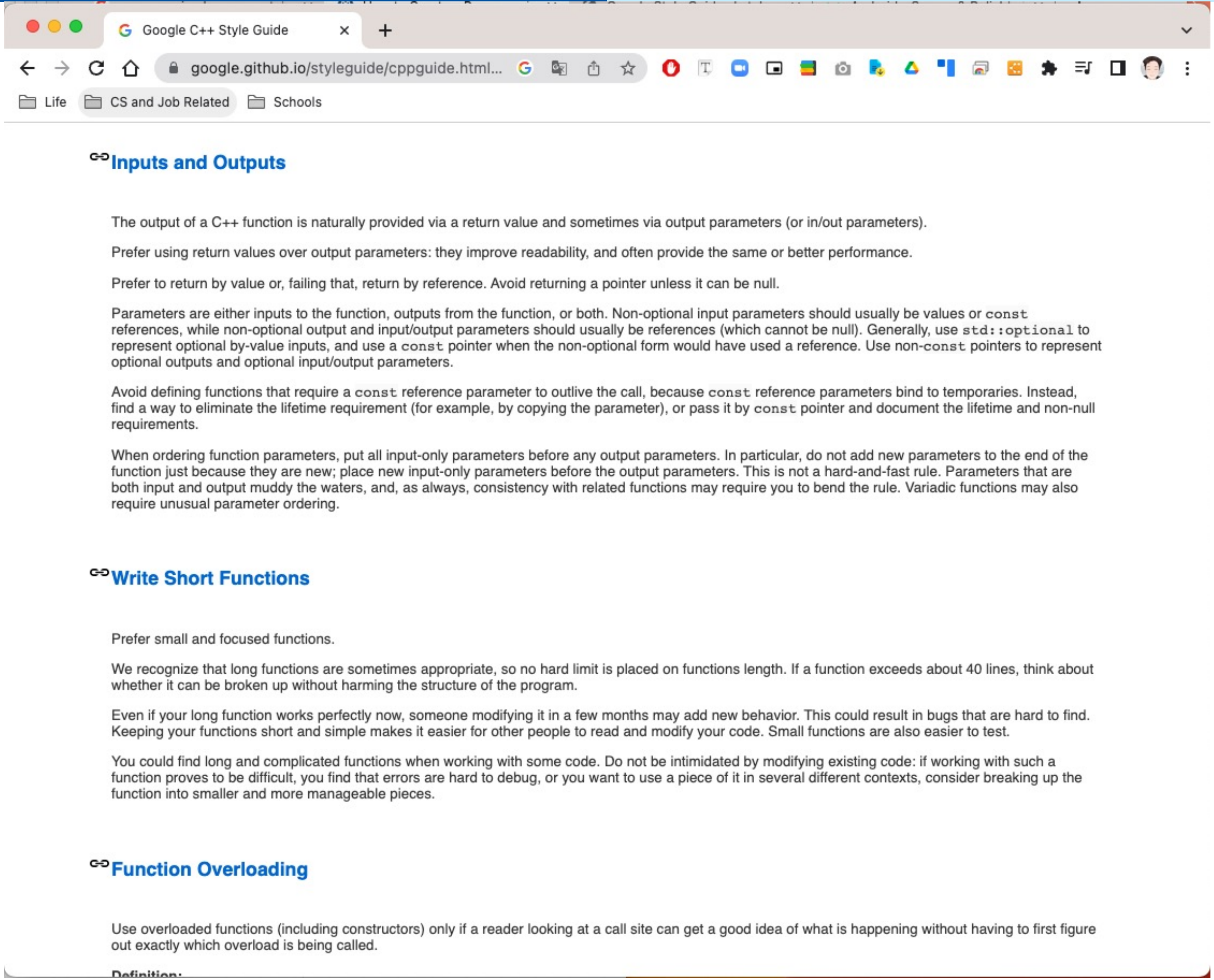
- **Purpose of code review - by numbers**
  - **From Steve McConnel's Code Complete**
  - Average defect detection rates
    - Unit testing: 25%
    - Function testing: 35%
    - Integration testing: 45%
    - Design and code inspections (reviews): 55% and 60%
  - 11 programs developed by the same group of people
    - First 5 without reviews: average 4.5 errors per 100 lines of code
    - Remaining 6 with reviews: average 0.82 errors per 100 lines of code
    - Errors reduced by > 80%
  - After AT&T introduced reviews, 14% increase in productivity and a 90% decrease in defects



Google C++ Style Guide

Table of Contents

C++ Version	
Header Files	Self-contained Headers The #define Guard Include What You Use Forward Declarations Inline Functions Names and Order of Includes
Scoping	Namespaces Internal Linkage Nonmember, Static Member, and Global Functions Local Variables Static and Global Variables thread_local Variables
Classes	Doing Work in Constructors Implicit Conversions Copyable and Movable Types Structs vs. Classes Structs vs. Pairs and Tuples Inheritance Operator Overloading Access Control Declaration Order
Functions	Inputs and Outputs Write Short Functions Function Overloading Default Arguments Trailing Return Type Syntax
Google-Specific Magic	Ownership and Smart Pointers cpplint
Other C++ Features	Rvalue References Friends Exceptions noexcept Run-Time Type Information (RTTI) Casting Streams Preincrement and Predecrement Use of const Use of constexpr Integer Types 64-bit Portability Preprocessor Macros 0 and nullptr/NULL sizeof Type Deduction (including auto) Class Template Argument Deduction Designated Initializers Lambda Expressions Template Metaprogramming Boost Other C++ Features Nonstandard Extensions Aliases
Inclusive Language	
Naming	General Naming Rules File Names Type Names Variable Names Constant Names Function Names Namespace Names Enumerator Names Macro Names Exceptions to Naming Rules
Comments	Comment Style File Comments Class Comments Function Comments Variable Comments Implementation Comments



Inputs and Outputs

The output of a C++ function is naturally provided via a return value and sometimes via output parameters (or in/out parameters).

Prefer using return values over output parameters: they improve readability, and often provide the same or better performance.

Prefer to return by value or, failing that, return by reference. Avoid returning a pointer unless it can be null.

Parameters are either inputs to the function, outputs from the function, or both. Non-optional input parameters should usually be values or `const` references, while non-optional output and input/output parameters should usually be references (which cannot be null). Generally, use `std::optional` to represent optional by-value inputs, and use a `const` pointer when the non-optional form would have used a reference. Use non-`const` pointers to represent optional outputs and optional input/output parameters.

Avoid defining functions that require a `const` reference parameter to outlive the call, because `const` reference parameters bind to temporaries. Instead, find a way to eliminate the lifetime requirement (for example, by copying the parameter), or pass it by `const` pointer and document the lifetime and non-null requirements.

When ordering function parameters, put all input-only parameters before any output parameters. In particular, do not add new parameters to the end of the function just because they are new; place new input-only parameters before the output parameters. This is not a hard-and-fast rule. Parameters that are both input and output muddy the waters, and, as always, consistency with related functions may require you to bend the rule. Variadic functions may also require unusual parameter ordering.

Write Short Functions

Prefer small and focused functions.

We recognize that long functions are sometimes appropriate, so no hard limit is placed on functions length. If a function exceeds about 40 lines, think about whether it can be broken up without harming the structure of the program.

Even if your long function works perfectly now, someone modifying it in a few months may add new behavior. This could result in bugs that are hard to find. Keeping your functions short and simple makes it easier for other people to read and modify your code. Small functions are also easier to test.

You could find long and complicated functions when working with some code. Do not be intimidated by modifying existing code: if working with such a function proves to be difficult, you find that errors are hard to debug, or you want to use a piece of it in several different contexts, consider breaking up the function into smaller and more manageable pieces.

Function Overloading

Use overloaded functions (including constructors) only if a reader looking at a call site can get a good idea of what is happening without having to first figure out exactly which overload is being called.

Definition:

Examples

- **Some common guidelines in multiple C++ style guide**
  - The rule of the three: If a class defines one (or more) of the following, it should explicitly define all three, which are 1) destructor, 2) copy constructor, 3) copy assignment operator
  - Do not use #define unless you have to use it
  - Try to use **const** member functions and variables
  - Set up the criteria on class, function, field, and variable names
  - Locate functions in proper classes
  - Try to use initializer list
  - Use iteration over STL containers
  - …

```
#define Fresh 1
#define Sophomore 2
#define Junior 3
#define Senior 4

class Student {
  public:
  Student(int id, int year) {
    student_id = id;
    student_year = year;
  };

  ~Student();

  int GetStudentID() { return student_id; }
  int get_student_year() { return student_year; }

  private:
  int student_id;
  int student_year;
};


bool FindStudent(int id, std::vector<Student> students) {
  for (int i = 0; i < students.size(); i++) {
    if (students[i].GetStudentID() == id) {
      return true;
    }
  }
  return false;
}
```

Do not use #define

Initializer list is not used

Violate the rule of the three

Inconsistency in function names

Fields are not distinguishable from local variables

References should be used

Iterator is not used

```cpp
class Student {
  public:
  enum StudentYear { FRESH = 1, Sophomore, Junior, Senior };

  Student(const int id, const StudentYear year) :
    id_(id), year_(year) {};
  Student(const Student& student) :
    id_(student.id_), year_(student.year_) {};
  Student& operator=(const Student& student) {
    if (this != &student) {
      *this = Student(student);
    }
    return *this;
  };
  ~Student();

  int GetId() { return id_; }
  int GetYear() { return year_; }

  bool FindStudent(const int id,
    const std::vector<Student>& students) const {
    for (const auto& student : students) {
      if (student.id_ == id) {
        return true;
      }
    }
    return false;
  }

  private:
  const int id_;
  StudentYear year_;
};
```

Google technologies: GoogleTest Framework

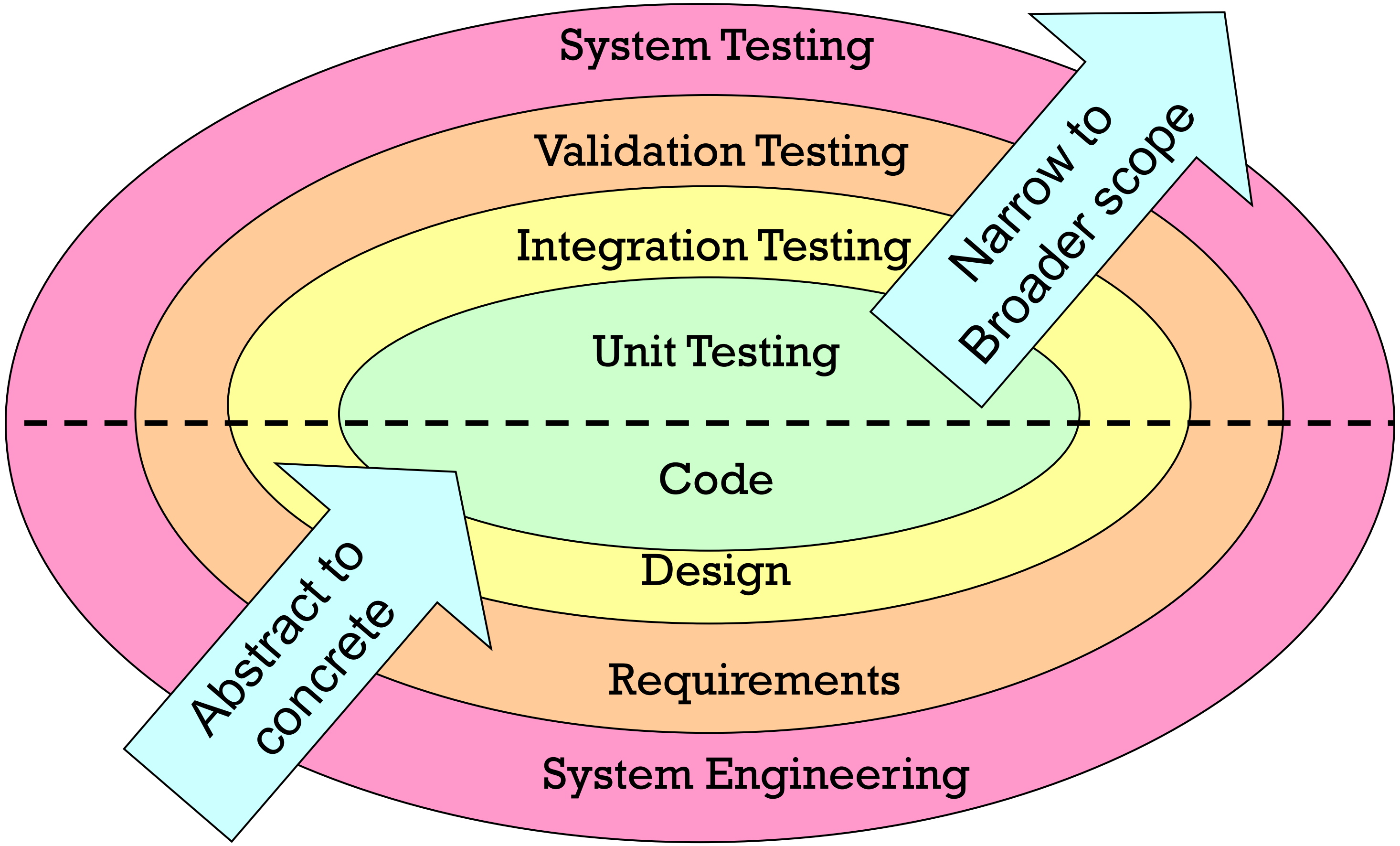
인하대학교
INHA UNIVERSITY

Software testing

- **Software testing**
  - Evaluation of the software against requirements gathered from users and system specifications
- **Does testing really work?**
  - "measuring over 20 projects: if you have a large number of unit tests your code will be **an order of magnitude (x10)** less complex.“
  - **Controlled study results**:
    - *“..quality increased linearly with the number of programmer tests...”*
    - *“..test-first students on average wrote more tests and, in turn, students who wrote more tests tended to be more productive...”*

http://agilepainrelief.com/notesfromatooluser/2008/11/misconceptions-with-test-driven-development.html
http://collaboration.csc.ncsu.edu/laurie/Papers/TDDpaperv8.pdf

Software testing



Software testing

- **Unit testing frameworks and libraries**
  - **Java**
    - NUnit, **Junit**, TestNG, Mockito, and PHPUnit
  - **Python**
    - Robot, **PyTest**, **Unittest**, DocTest, Nose2, and Testify
  - **C/C++**
    - **Googletest**, Boot Test Library, QA Systems Cantata, Parasoft C/C++ test, Microsoft Visual Studio, Cppunit, Catch, Bandit, and CppUTest
  - **JavaScript**
    - Jest, Mocah, Storybook, Jasmine, Cypress, Puppeteer, Testing Library, and WebdriverIO

인하대학교
INHA UNIVERSITY

Googletest framework

- **Googletest framework**
  - A **unit testing library** for the C++ programming language.
  - **Repository**
    - http://code.google.com/p/googletest/
  - **Projects using Google Test**
    - Android open source project operating system
    - Chromium projects (behind the Chrome browser, Edge browser, and Chrome OS)
    - LLVM compiler
    - Protocol Buffers (Google's data interchange format)
    - OpenCV computer vision library
    - Several internal C++ projects at Google
  - **Study materials**
    - README file: https://github.com/google/googletest/blob/master/README.md
    - Googletest user's guide: https://google.github.io/googletest/
    - Whittaker, James (2012). How Google Tests Software. Boston, Massachusetts: Pearson Education. ISBN 0-321-80302-7

Create tests

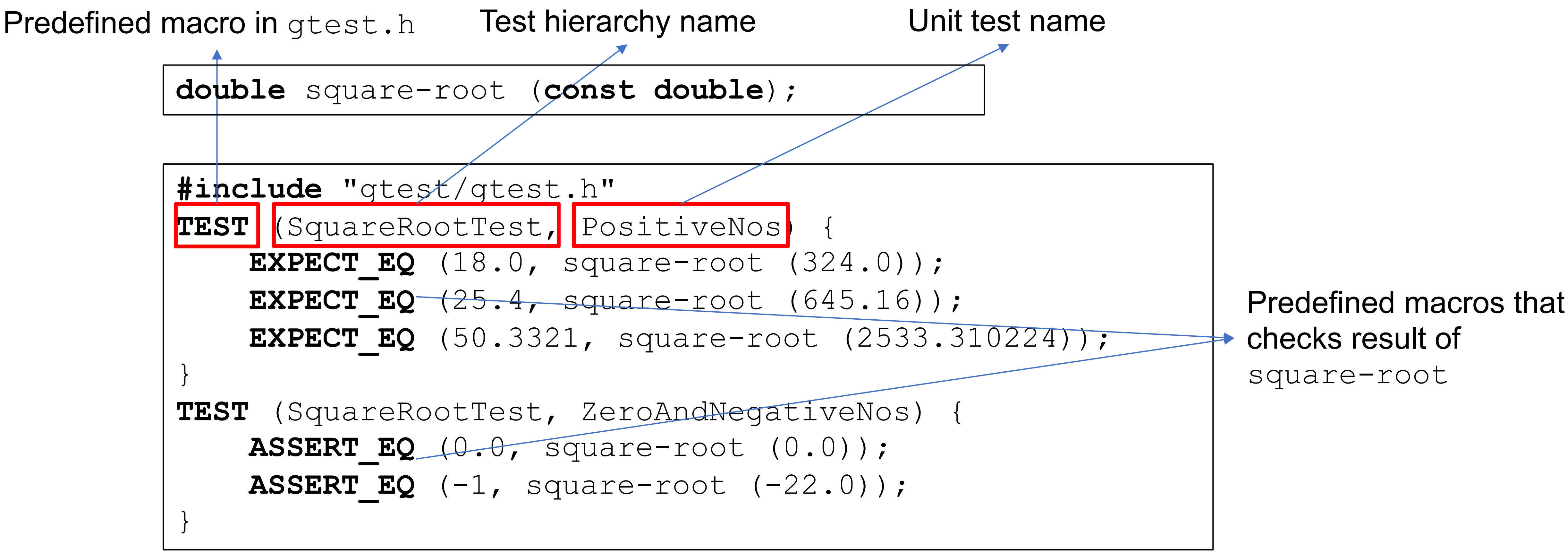
- **Creating a basic test**
  - **Target code: prototype for square-root**

```
double square-root (const double);
```

  - **Test case with Googletest**

```
#include "gtest/gtest.h"
TEST (SquareRootTest, PositiveNos) {
    EXPECT_EQ (18.0, square-root (324.0));
    EXPECT_EQ (25.4, square-root (645.16));
    EXPECT_EQ (50.3321, square-root (2533.310224));
}
TEST (SquareRootTest, ZeroAndNegativeNos) {
    ASSERT_EQ (0.0, square-root (0.0));
    ASSERT_EQ (-1, square-root (-22.0));
}
```

Create tests



Predefined macro in `gtest.h`

Test hierarchy name

Unit test name

```
double square-root (const double);
```

```
#include "gtest/gtest.h"
TEST (SquareRootTest, PositiveNos) {
    EXPECT_EQ (18.0, square-root (324.0));
    EXPECT_EQ (25.4, square-root (645.16));
    EXPECT_EQ (50.3321, square-root (2533.310224));
}
TEST (SquareRootTest, ZeroAndNegativeNos) {
    ASSERT_EQ (0.0, square-root (0.0));
    ASSERT_EQ (-1, square-root (-22.0));
}
```

Predefined macros that checks result of `square-root`

Check results

- Basic assertions

Fatal assertion	Nonfatal assertion	Verifies
`ASSERT_TRUE(condtion);`	`EXPECT_TRUE(condtion);`	`Condition` **is true**
`ASSERT_FALSE(condition);`	`EXPECT_FALSE(condition);`	`Condition` **is false**

Check results

- Binary comparison

Fatal assertion	Nonfatal assertion	Verifies
`ASSERT_EQ(expected, actual);`	`EXPECTED_EQ(expected, actual);`	`expected == actual`
`ASSERT_NE(val1, val2);`	`EXPECT_NE(val1, val2);`	`val1 != val2`
`ASSERT_LT(val1, val2);`	`EXPECT_LT(val1, val2);`	`val1 < val2`
`ASSERT_LE(val1, val2);`	`EXPECT_LE(val1, val2);`	`val1 <= val2`
`ASSERT_GT(val1, val2);`	`EXPECT_GT(val1, val2);`	`val1 > val2`
`ASSERT_GE(val1, val2);`	`EXPECT_GE(val1, val2);`	`val1 >= val2`

Run tests

- Initialize the framework
- Must be called before `RUN_ALL_TESTS`

```
int main(int argc, char **argv) {
    ::testing::InitGoogleTest(&argc, argv);
    return RUN_ALL_TESTS();
}
```

- Must be called only once
  - Multiple calls to it conflicts some features of the framework
- Automatically detects and runs all test tests defined using the `TEST` macro

Run tests

```
Running main() from user_main.cpp
[==========] Running 2 tests from 1 test case.
[----------] Global test environment set-up.
[----------] 2 tests from SquareRootTest
[ RUN      ] SquareRootTest.PositiveNos
..\user_sqrt.cpp(6862): error: Value of: sqrt (2533.310224)
  Actual: 50.332
Expected: 50.3321
[  FAILED  ] SquareRootTest.PositiveNos (9 ms)
[ RUN      ] SquareRootTest.ZeroAndNegativeNos
[       OK ] SquareRootTest.ZeroAndNegativeNos (0 ms)
[----------] 2 tests from SquareRootTest (0 ms total)

[----------] Global test environment tear-down
[==========] 2 tests from 1 test case ran. (10 ms total)
[  PASSED  ] 1 test.
[  FAILED  ] 1 test, listed below:
[  FAILED  ] SquareRootTest.PositiveNos

1 FAILED TEST
```

Google technologies:
Machine learning related tools

인하대학교
INHA UNIVERSITY

Machine learning related tools

- **Machine learning related tools**
  - Colab: Allows you to write and execute Python in your browser, with zero configuration required, access to GPUs free of charge, and easy sharing
  - Tensorflow and Keras
    - Tensorflow : A free and open-source software library for machine learning and artificial intelligence
    - Keras (an interface for the Tensorflow library): An open-source software library that provides a Python interface for artificial neural networks
  - Tensorflow hub and model garden: Pre-built tensorflow models
  - Tensorflow model optimization toolkit: A suite of tools for optimizing ML models for deployment and execution
  - Tensorboard: Provides the visualization and tooling needed for machine learning experimentation

Colab

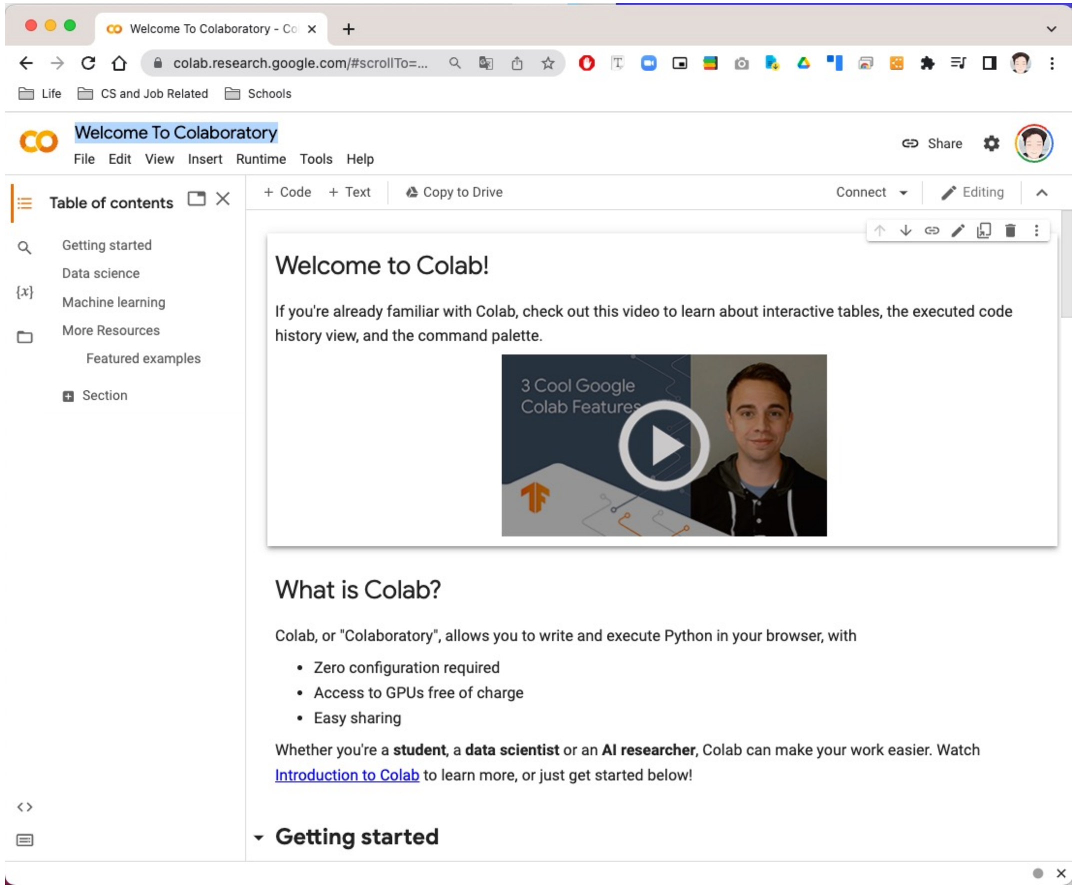
- **Colab**
  - A project from Google Research, a free, Jupyter based environment
  - It allows us to create Jupyter programming notebooks to write and execute Python in a web browser
    - It also supports other Python-based third-party tools and machine learning frameworks such as Pandas, PyTorch, Tensorflow, Keras, Monk, OpenCV, and others
  - Google provides the use of free GPU for your Colab notebooks

Colab

- **Introduction to Colab**

Colab

- **Welcome To Colab (https://colab.research.google.com/)**



Welcome to Colab!

If you're already familiar with Colab, check out this video to learn about interactive tables, the executed code history view, and the command palette.

What is Colab?

Colab, or "Colaboratory", allows you to write and execute Python in your browser, with

- Zero configuration required
- Access to GPUs free of charge
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch Introduction to Colab to learn more, or just get started below!

Getting started

인하대학교 INHA UNIVERSITY

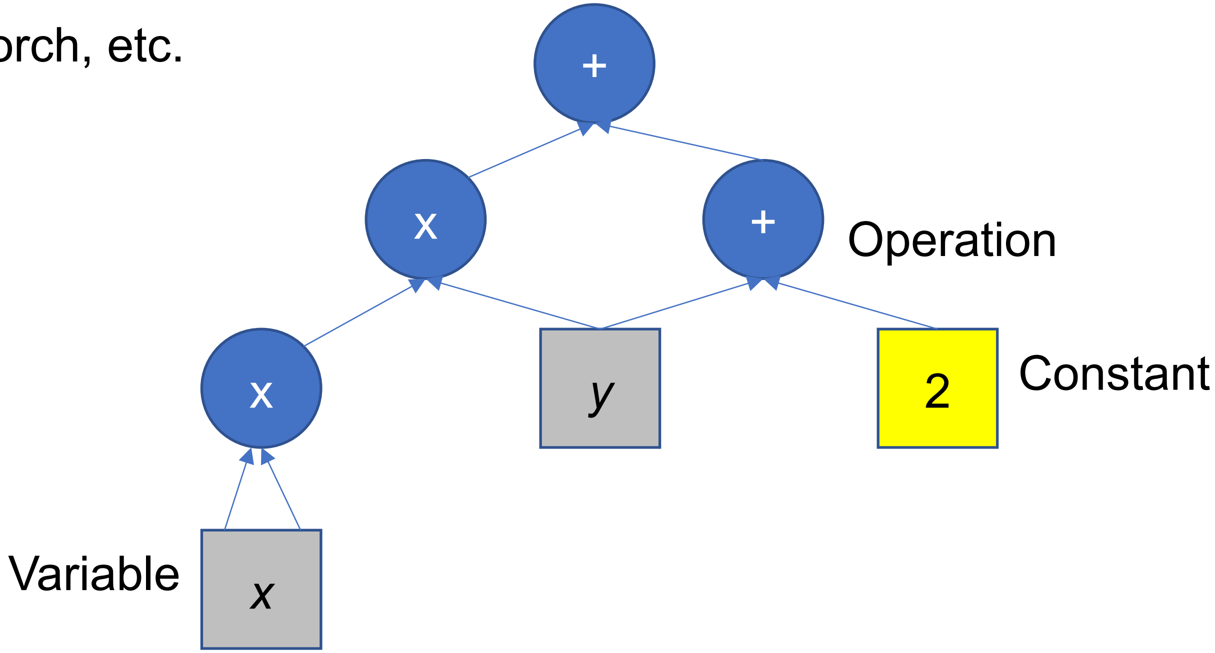
Tensorflow and Keras

- **Deep learning frameworks**

Caffe
torch
Chainer
PyTorch
DEEPLEARNING4J
TensorFlow
Caffe2
The Microsoft Cognitive Toolkit
K
theano
dmlc
mxnet

Tensorflow and Keras

- **Deep learning frameworks**
  - Tools for defining static or dynamic general-purpose computational graphs
  - Seamless CPU / GPU usage
  - Python / numpy or R interfaces instead of C, C++, CUDA or HIP
  - Open source
  - Tensorflow, PyTorch, etc.



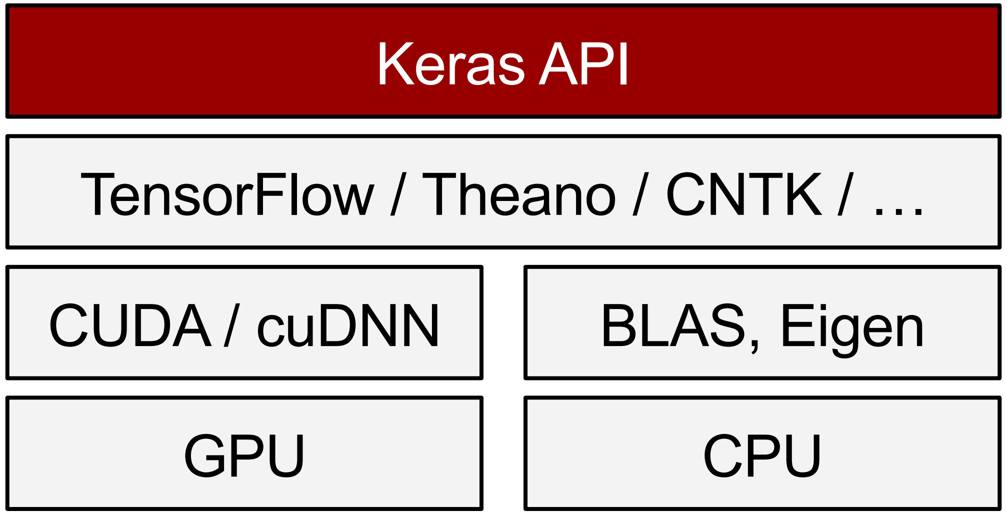
Tensorflow and Keras

- **Tensorflow (https://www.tensorflow.org/)**
  - Deep learning frameworks from Google (The initial version was the name with DistBelief)
  - It can be used in a wide variety of programming languages (e.g., Python, JavaScript, C++, and Java)
  - Companies using Tensorflow
    - Google : Translate, Google Brain's Magenta
    - DeepMind : WaveNet Text to Speech

airbnb
AMD
NVIDIA.
UBER
SAP
kakao
DeepMind
Dropbox
ebay
Google
intel
Coca-Cola
mi
QUALCOMM

Tensorflow and Keras

- **Keras**
  - A high-level neural networks API
  - A frontend API for Tensorflow 2.0
    - https://keras.io/
    - https://www.tensorflow.org/guide/keras
  - It dramatically increase the usability of Tensorflow



Keras API

TensorFlow / Theano / CNTK / …

CUDA / cuDNN

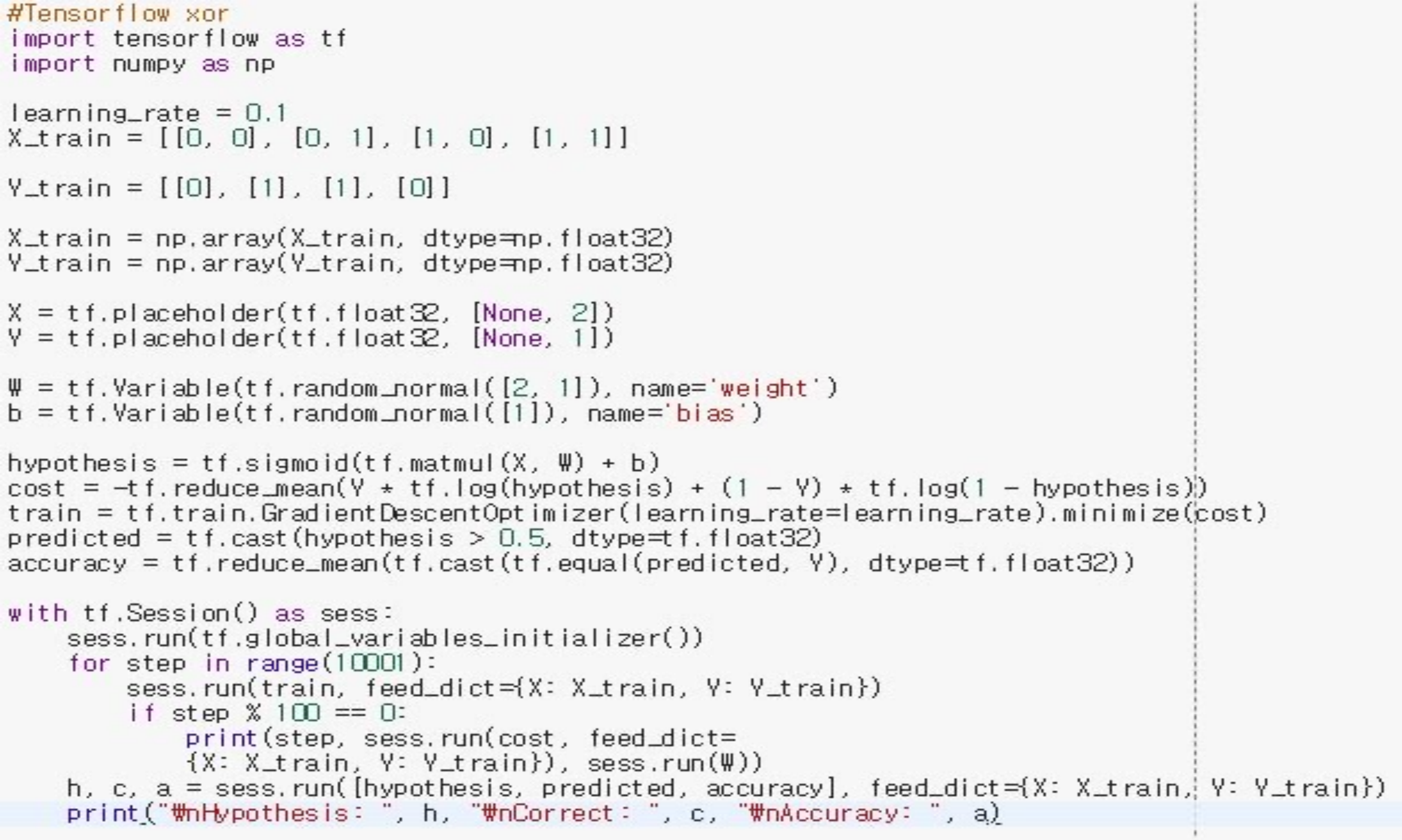
BLAS, Eigen

GPU

CPU

Tensorflow and Keras

- **Tensorflow 1.0 and Tensorflow 2.0**

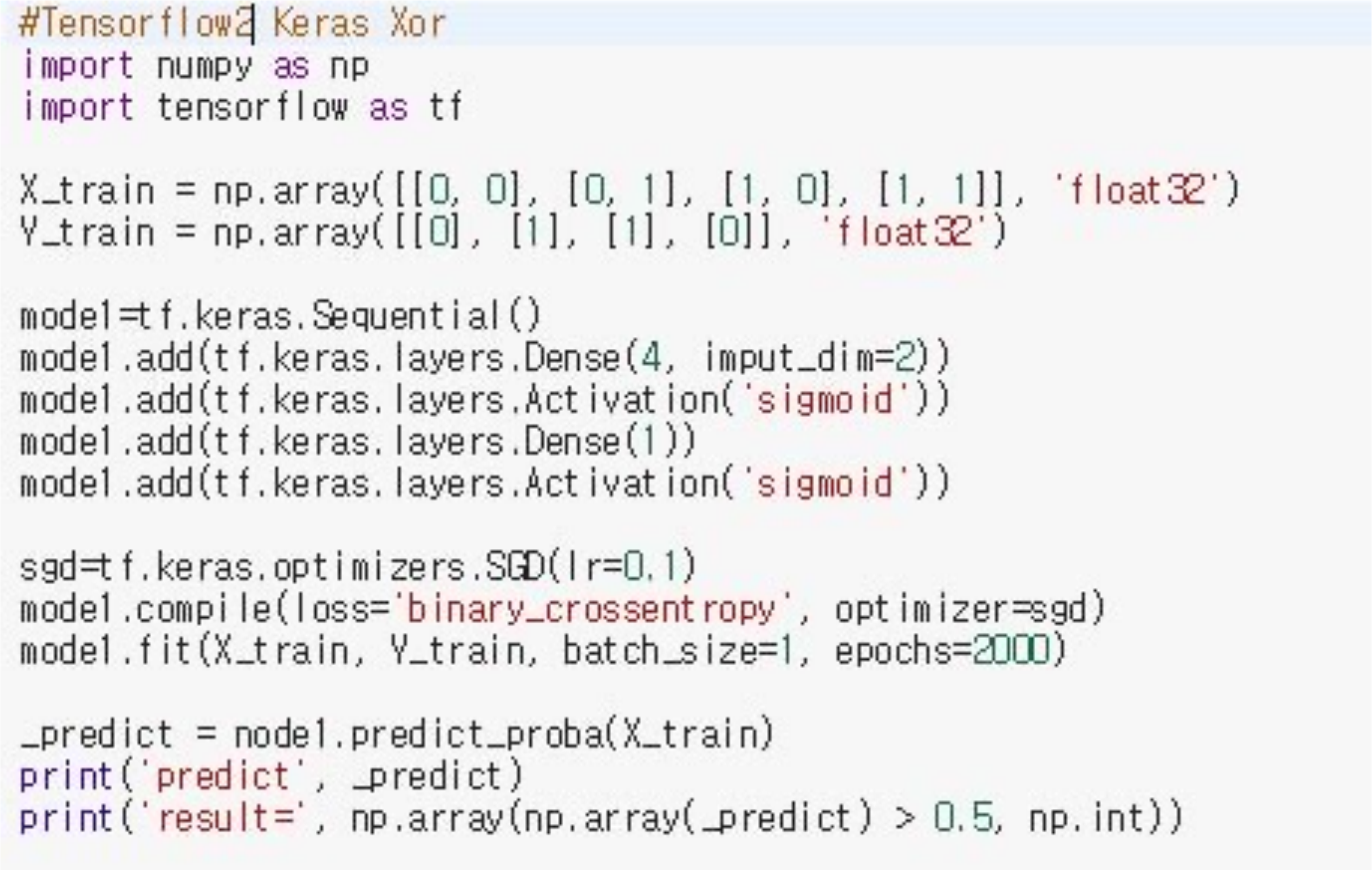


```
#Tensorflow xor
import tensorflow as tf
import numpy as np

learning_rate = 0.1
X_train = [[0, 0], [0, 1], [1, 0], [1, 1]]

Y_train = [[0], [1], [1], [0]]

X_train = np.array(X_train, dtype=np.float32)
Y_train = np.array(Y_train, dtype=np.float32)

X = tf.placeholder(tf.float32, [None, 2])
Y = tf.placeholder(tf.float32, [None, 1])

W = tf.Variable(tf.random_normal([2, 1]), name='weight')
b = tf.Variable(tf.random_normal([1]), name='bias')

hypothesis = tf.sigmoid(tf.matmul(X, W) + b)
cost = -tf.reduce_mean(Y * tf.log(hypothesis) + (1 - Y) * tf.log(1 - hypothesis))
train = tf.train.GradientDescentOptimizer(learning_rate=learning_rate).minimize(cost)
predicted = tf.cast(hypothesis > 0.5, dtype=tf.float32)
accuracy = tf.reduce_mean(tf.cast(tf.equal(predicted, Y), dtype=tf.float32))

with tf.Session() as sess:
    sess.run(tf.global_variables_initializer())
    for step in range(10001):
        sess.run(train, feed_dict={X: X_train, Y: Y_train})
        if step % 100 == 0:
            print(step, sess.run(cost, feed_dict=
            {X: X_train, Y: Y_train}), sess.run(W))
    h, c, a = sess.run([hypothesis, predicted, accuracy], feed_dict={X: X_train, Y: Y_train})
    print("₩nHypothesis: ", h, "₩nCorrect: ", c, "₩nAccuracy: ", a)
```

인하대학교
INHA UNIVERSITY

Tensorflow and Keras

- **Tensorflow 1.0 and Tensorflow 2.0**

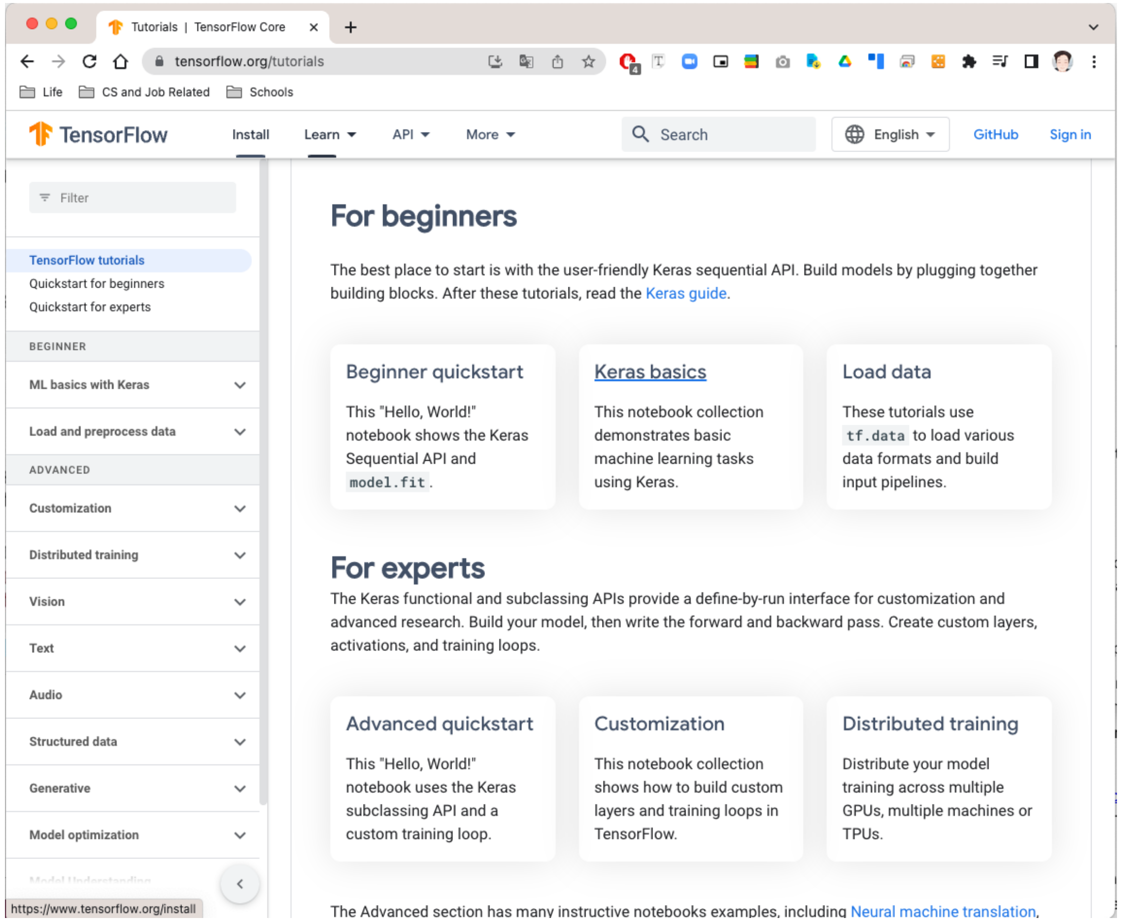


```
#Tensorflow2 Keras Xor
import numpy as np
import tensorflow as tf

X_train = np.array([[0, 0], [0, 1], [1, 0], [1, 1]], 'float32')
Y_train = np.array([[0], [1], [1], [0]], 'float32')

mode1=tf.keras.Sequential()
mode1.add(tf.keras.layers.Dense(4, imput_dim=2))
mode1.add(tf.keras.layers.Activation('sigmoid'))
mode1.add(tf.keras.layers.Dense(1))
mode1.add(tf.keras.layers.Activation('sigmoid'))

sgd=tf.keras.optimizers.SGD(lr=0.1)
mode1.compile(loss='binary_crossentropy', optimizer=sgd)
mode1.fit(X_train, Y_train, batch_size=1, epochs=2000)

_predict = node1.predict_proba(X_train)
print('predict', _predict)
print('result=', np.array(np.array(_predict) > 0.5, np.int))
```

인하대학교
INHA UNIVERSITY

Tensorflow and Keras

https://www.tensorflow.org/tutorials



For beginners

The best place to start is with the user-friendly Keras sequential API. Build models by plugging together building blocks. After these tutorials, read the Keras guide.

Beginner quickstart

This "Hello, World!" notebook shows the Keras Sequential API and `model.fit`.

Keras basics

This notebook collection demonstrates basic machine learning tasks using Keras.

Load data

These tutorials use `tf.data` to load various data formats and build input pipelines.

For experts

The Keras functional and subclassing APIs provide a define-by-run interface for customization and advanced research. Build your model, then write the forward and backward pass. Create custom layers, activations, and training loops.

Advanced quickstart

This "Hello, World!" notebook uses the Keras subclassing API and a custom training loop.

Customization

This notebook collection shows how to build custom layers and training loops in TensorFlow.

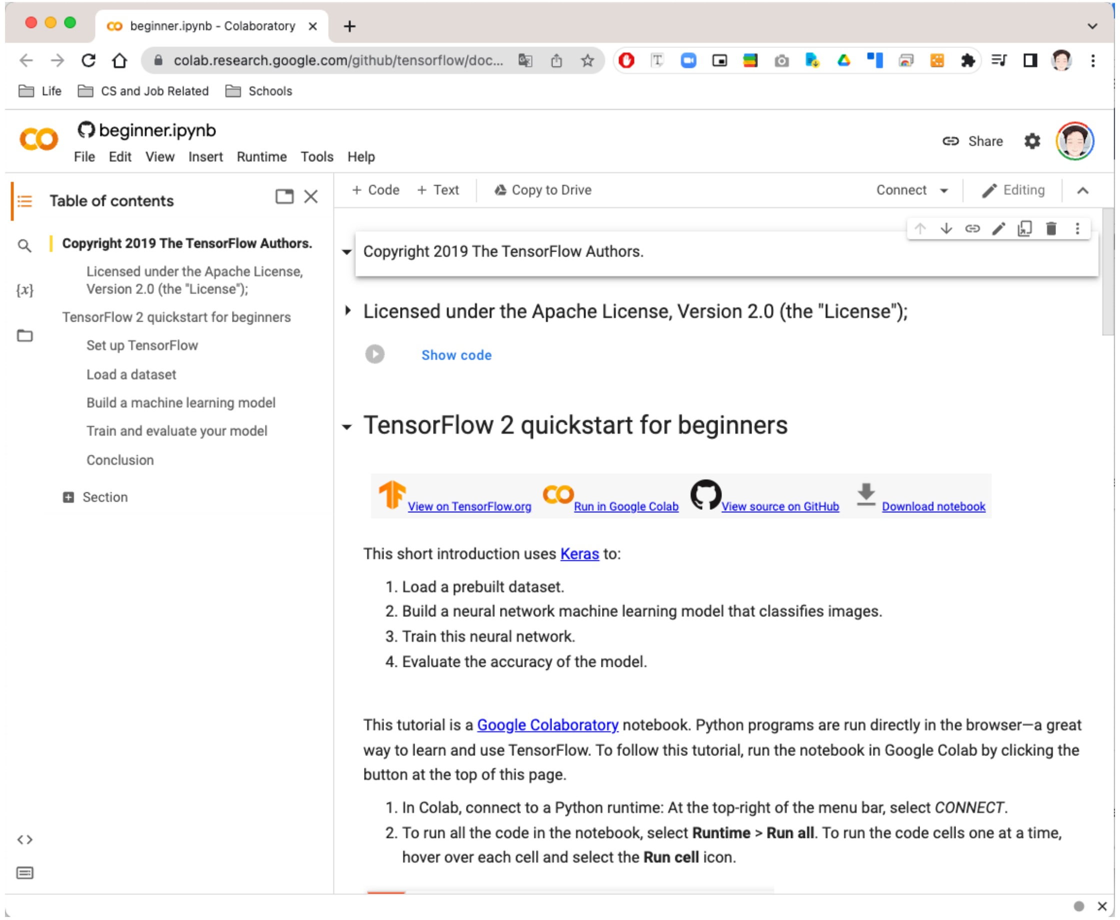
Distributed training

Distribute your model training across multiple GPUs, multiple machines or TPUs.

The Advanced section has many instructive notebooks examples, including Neural machine translation,

Tensorflow and Keras

https://www.tensorflow.org/tutorials/quickstart/beginner



- Copyright 2019 The TensorFlow Authors.
  - Licensed under the Apache License, Version 2.0 (the "License");
- TensorFlow 2 quickstart for beginners
  - Set up TensorFlow
  - Load a dataset
  - Build a machine learning model
  - Train and evaluate your model
  - Conclusion

Copyright 2019 The TensorFlow Authors.

Licensed under the Apache License, Version 2.0 (the "License");

TensorFlow 2 quickstart for beginners

This short introduction uses Keras to:

1. Load a prebuilt dataset.
2. Build a neural network machine learning model that classifies images.
3. Train this neural network.
4. Evaluate the accuracy of the model.

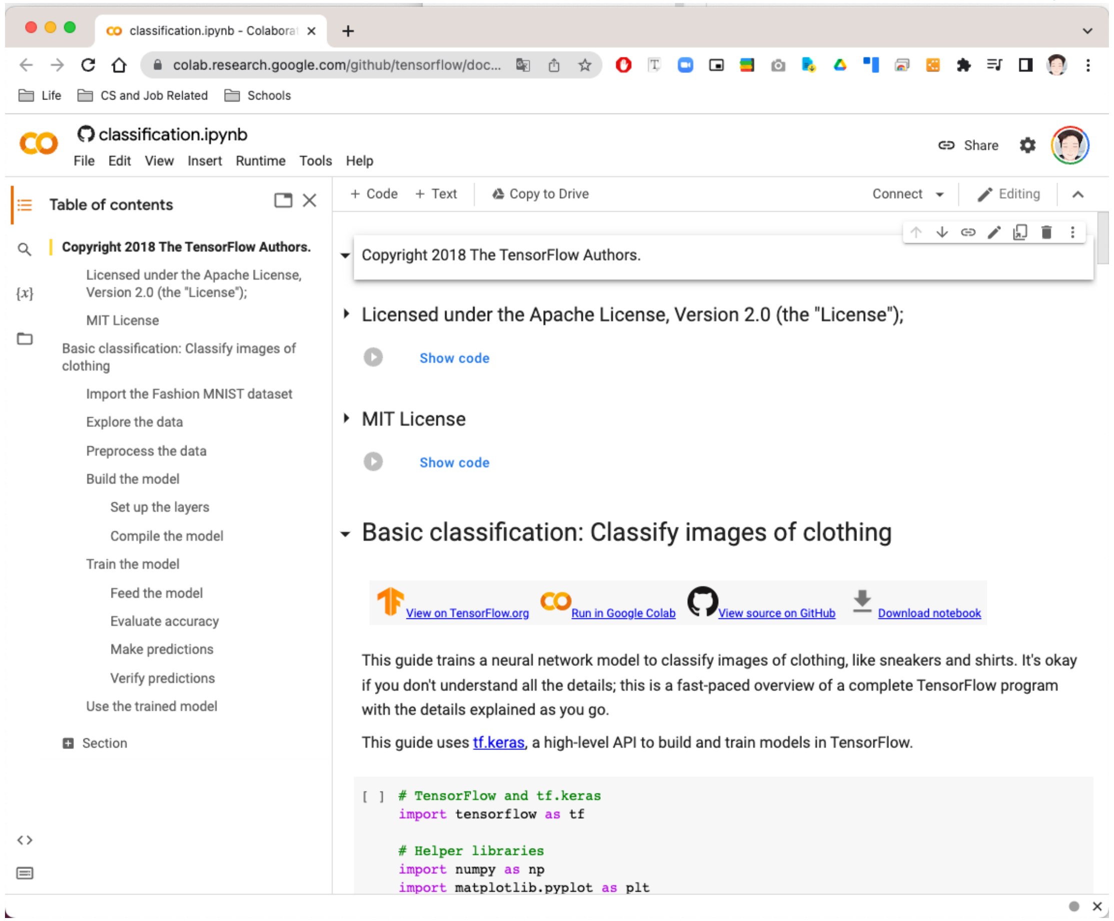
This tutorial is a Google Colaboratory notebook. Python programs are run directly in the browser—a great way to learn and use TensorFlow. To follow this tutorial, run the notebook in Google Colab by clicking the button at the top of this page.

1. In Colab, connect to a Python runtime: At the top-right of the menu bar, select *CONNECT*.
2. To run all the code in the notebook, select **Runtime** > **Run all**. To run the code cells one at a time, hover over each cell and select the **Run cell** icon.

인하대학교 INHA UNIVERSITY

Tensorflow and Keras

https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/keras/classification.ipynb



classification.ipynb - Colaborat
colab.research.google.com/github/tensorflow/doc...
Life
CS and Job Related
Schools
classification.ipynb
File Edit View Insert Runtime Tools Help
Share
Table of contents
+ Code + Text
Copy to Drive
Connect
Editing

- Copyright 2018 The TensorFlow Authors.
  - Licensed under the Apache License, Version 2.0 (the "License");
  - MIT License
- Basic classification: Classify images of clothing
  - Import the Fashion MNIST dataset
  - Explore the data
  - Preprocess the data
  - Build the model
    - Set up the layers
    - Compile the model
  - Train the model
    - Feed the model
    - Evaluate accuracy
    - Make predictions
    - Verify predictions
  - Use the trained model

Section

Copyright 2018 The TensorFlow Authors.

Licensed under the Apache License, Version 2.0 (the "License");

Show code

MIT License

Show code

Basic classification: Classify images of clothing

View on TensorFlow.org | Run in Google Colab | View source on GitHub | Download notebook

This guide trains a neural network model to classify images of clothing, like sneakers and shirts. It's okay if you don't understand all the details; this is a fast-paced overview of a complete TensorFlow program with the details explained as you go.

This guide uses tf.keras, a high-level API to build and train models in TensorFlow.

```
[ ] # TensorFlow and tf.keras
    import tensorflow as tf

    # Helper libraries
    import numpy as np
    import matplotlib.pyplot as plt
```

Tensorflow and Keras

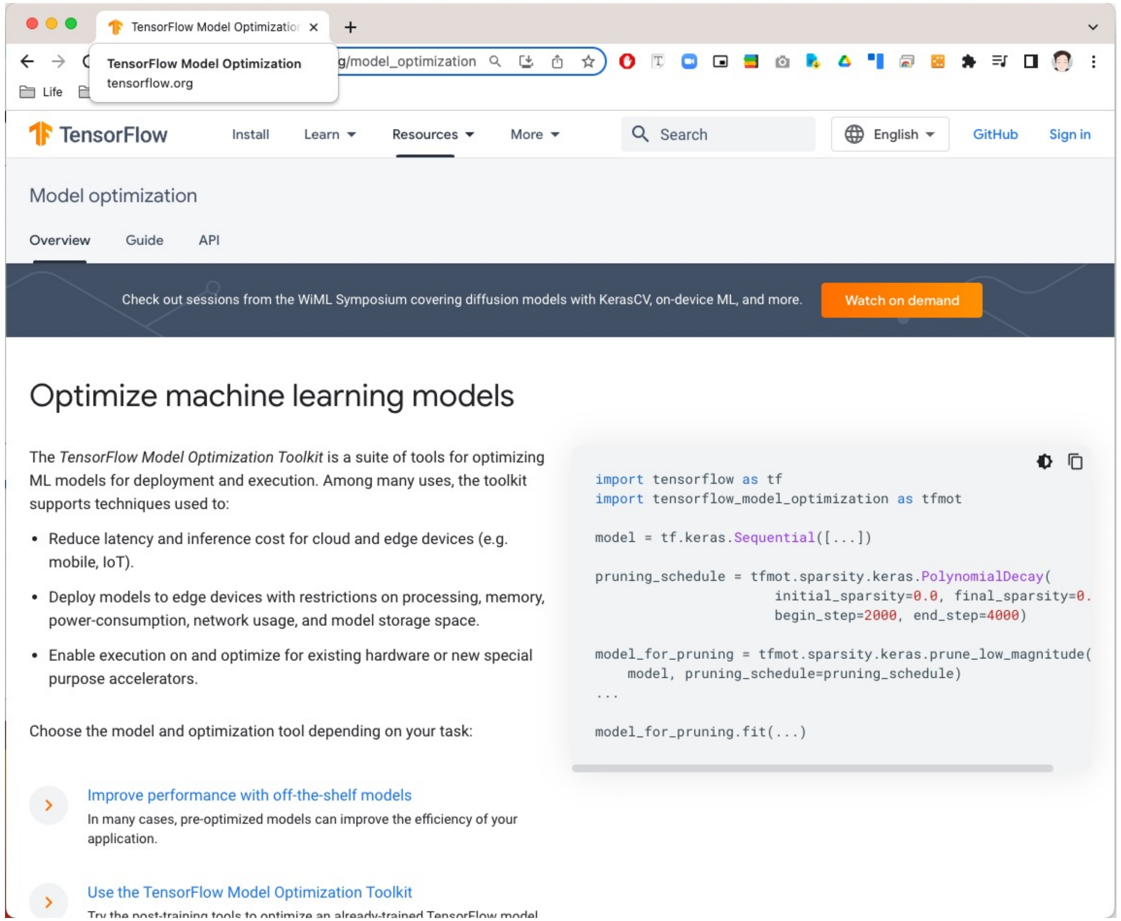
- **Tensorflow hub and model garden**
  - Tensorflow Hub (https://www.tensorflow.org/hub)
    - A repository of trained machine learning models ready for fine-tuning and deployable anywhere
    - Reuse trained models like BERT and Faster R-CNN with just a few lines of code
  - Model garden (https://www.tensorflow.org/guide/model_garden)
    - Provides implementations of many state-of-the-art machine learning (ML) models for vision and natural language processing (NLP)
    - Provides workflow tools to let you **quickly configure** and run those models on **standard datasets**

Tensorflow model optimization toolkit

- **Optimize machine learning models**
  - A suite of tools for optimizing ML models for deployment and execution
  - The toolkit supports techniques used to:
    - Reduce latency and inference cost for cloud and edge devices (e.g., mobile, IoT)
    - Deploy models to edge devices with restrictions on processing, memory, power-consumption, network usage, and model storage space
    - Enable execution on and optimize for existing hardware or new special purpose accelerators
  - Two famous techniques
    - Weight pruning
    - **Quantization**
    - Weight clustering

Tensorflow model optimization toolkit

https://www.tensorflow.org/model_optimization



Model optimization

Check out sessions from the WiML Symposium covering diffusion models with KerasCV, on-device ML, and more. Watch on demand

Optimize machine learning models

The *TensorFlow Model Optimization Toolkit* is a suite of tools for optimizing ML models for deployment and execution. Among many uses, the toolkit supports techniques used to:

- Reduce latency and inference cost for cloud and edge devices (e.g. mobile, IoT).
- Deploy models to edge devices with restrictions on processing, memory, power-consumption, network usage, and model storage space.
- Enable execution on and optimize for existing hardware or new special purpose accelerators.

Choose the model and optimization tool depending on your task:

```
import tensorflow as tf
import tensorflow_model_optimization as tfmot

model = tf.keras.Sequential([...])

pruning_schedule = tfmot.sparsity.keras.PolynomialDecay(
                      initial_sparsity=0.0, final_sparsity=0.
                      begin_step=2000, end_step=4000)

model_for_pruning = tfmot.sparsity.keras.prune_low_magnitude(
    model, pruning_schedule=pruning_schedule)
...

model_for_pruning.fit(...)
```

Improve performance with off-the-shelf models

In many cases, pre-optimized models can improve the efficiency of your application.

Use the TensorFlow Model Optimization Toolkit

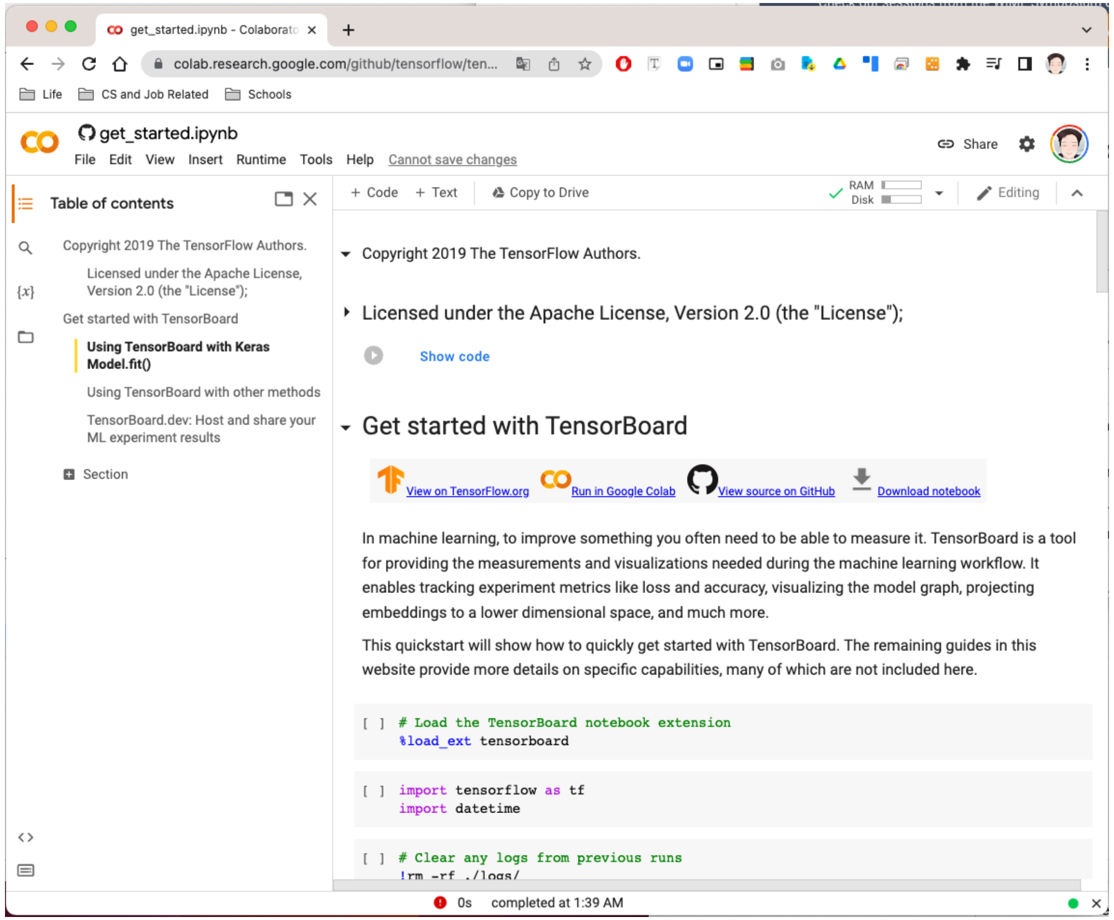
Try the post-training tools to optimize an already-trained TensorFlow model

Tensorboard

- **Tensorboard (https://www.tensorflow.org/tensorboard)**
  - provides the visualization and tooling needed for machine learning experimentation:
    - Tracking and visualizing metrics such as loss and accuracy
    - Visualizing the model graph (ops and layers)
    - Viewing histograms of weights, biases, or other tensors as they change over time
    - Projecting embeddings to a lower dimensional space
    - Displaying images, text, and audio data
    - Profiling Tensorflow programs
    - And much more

Tensorboard

https://www.tensorflow.org/tensorboard/get_started


get_started.ipynb - Colaborato
colab.research.google.com/github/tensorflow/ten...
Life
CS and Job Related
Schools
get_started.ipynb
File
Edit
View
Insert
Runtime
Tools
Help
Cannot save changes
Share
+ Code
+ Text
Copy to Drive
RAM
Disk
Editing
Table of contents
Copyright 2019 The TensorFlow Authors.
Licensed under the Apache License, Version 2.0 (the "License");
Get started with TensorBoard
Using TensorBoard with Keras Model.fit()
Using TensorBoard with other methods
TensorBoard.dev: Host and share your ML experiment results
Section
Show code
View on TensorFlow.org
Run in Google Colab
View source on GitHub
Download notebook
0s
completed at 1:39 AM

Copyright 2019 The TensorFlow Authors.

Licensed under the Apache License, Version 2.0 (the "License");

Get started with TensorBoard

In machine learning, to improve something you often need to be able to measure it. TensorBoard is a tool for providing the measurements and visualizations needed during the machine learning workflow. It enables tracking experiment metrics like loss and accuracy, visualizing the model graph, projecting embeddings to a lower dimensional space, and much more.

This quickstart will show how to quickly get started with TensorBoard. The remaining guides in this website provide more details on specific capabilities, many of which are not included here.

```
[ ]  # Load the TensorBoard notebook extension
     %load_ext tensorboard
```

```
[ ]  import tensorflow as tf
     import datetime
```

```
[ ]  # Clear any logs from previous runs
     !rm -rf ./logs/
```

Summary

인하대학교
INHA UNIVERSITY

Summary

- **Summary**
  - Websites and Youtube channels for Google techs and lives
    - Provides a lot of useful information for Google and google techs
  - Programming language style guides and Googletest framework
    - Following the style guide and use a Googletest framework (only for C++) will dramatically improve the project quality
  - Machine learning related tools
    - Colab: A web-based and highly readable Python programming IDE
    - Tensorflow and Keras: A tool to build machine learning models
    - Tensorflow model optimization toolkit: A tool to optimize machine learning models
    - Tensorflow hub and model garden: Pre-built machine learning models
    - Tensorboard: A tool to experiment machine learning models

Q & A

인하대학교
INHA UNIVERSITY