

Competition Roadmap

This document is for potential competition organizers. It provides an overview of how competitions work, and describes the processes involved in creating and running CodaLab competitions. Beginners might want to use the [ChaLab](#) wizard.

- [How Competitions Work](#)
 - [Competition Bundle File Structure](#)
 - [Types of Competition](#)
- [Competition End to End Process](#)
 - [Planning a Competition](#)
 - [How submissions are processed](#)
 - [Making Data Available](#)
 - [Creating an Ingestion Program](#)
 - [Creating a Scoring Program](#)
 - [Creating a Starting Kit](#)
 - [Creating a Competition Bundle](#)
 - [Running a Competition](#)

How Competitions Work

In a typical CodaLab competition, participants compete to find the best approach for a particular problem. Competitions may be conducted in multiple phases, e.g. development, feedback, and final. The appropriate data is made available to participants at each phase of the competition. During the development and feedback phases, participants have access to training data to develop and refine their algorithms. During the final competition phase, participants are provided with final test data to generate results, which they can then submit to the competition. Results are calculated at the end of each phase, at which point participants can see the competition results on the leaderboard.

Competition Bundle File Structure

Competitions consist of a set of files collectively known as a "bundle". Although technically CodaLab considers any zipped archive to be a bundle, competition bundles generally contain a specific assortment of files:

- `competition.yaml`: Configuration file defining all the features of the competition, and linking to resources needed to organize the competition (html files, data, programs).
- HTML pages: Descriptive text and instructions to participants.
- program files: Files that make up the ingestion program, the scoring program, and the starting kit. With the default compute worker and docker on the public instance, we support Python (.py) files and Windows-based binary executables. Via the use of your own docker images and running your own compute workers, you have the flexibility of using any programming language and OS.
- data files: Contain training data and reference data for the competition.

This listing gives a general idea of the files which make up a competition, although this can vary depending on the type of competition. Beginners may want to use ChaLab, a [wizard](#) that guides you step-by-step to build a competition bundle.

Types of Competition

CodaLab competitions can be set up in several different ways. We provide several templates, which differ in the type of data and complexity of the organization. These three templates are the most basic:

- [Yellow World](#): The simplest template with RESULT submission. Competitors submit an answer which is compared with the known correct answer.
- [Compute Pi](#): The simplest template with CODE submission. Competitors submit an answer which is compared with the known correct answer.
- [Iris](#): A two-phase competition similar to what the [ChaLab wizard](#) can produce, with either result or code submission.

[More complex examples](#) are also provided for developers and advanced users.

Competition End to End Process

In this section we'll walk through the major segments of the competition creation process.

Planning a Competition

The very first step in creating a competition is planning. For help, see the [ChaLearn tips](#). It is advisable to first start preparing data, a scoring program, and a sample solution. This will be the basis for the "starting kit", which you can distribute to participants to get started. New in CodaLab 1.5: the "ingestion program", which allows you to parse the submission of participants.

How submissions are processed

The diagram below illustrates how submissions made by participants are processed. They are received by an ingestion program, which decides whether to treat them as "result" submission or "code" submission. Result submissions are directly forwarded to the scoring program whereas code submissions are executed on the platform to produce results. The scoring program compares the results with the solution (known only to the organizers).

[blocked URL](#)

Making Data Available

There are several kinds of data an organizer can provide in each phase:

- **Public data:** Data available for download from the competition page. Typically this can be `sample` data in a code submission competition or `labeled training` data and `unlabeled test` data in a result submission competition.
- **Input data:** Data used as input to the participants' code, in a code submission competition.
- **Reference data:** The "solution" of the problem, visible only to the scoring program.

The data format can freely be chosen by the organizers.

Codalab lets you either include data in your competition bundle (as zip files) or upload data in `My Competitions>My Datasets`, then reference the datasets from your YAML configuration file. You can also switch dataset on-the-fly while your competition is running by re-uploading a dataset and select the new version via the competition editor. See [My Datasets](#).

Creating an Ingestion Program

The ingestion program receives the submission of participants. You do not necessarily need to write one. Codalab provides a default ingestion program with the following behavior, depending on the type of submission made by the participant:

- If the submission contains no "metadata" file: treat it as a "result" submission and forward it directly to the "scoring program".
- If the submission contains a "metadata" file: execute the command found in the metadata file.

Each phase may have a different ingestion program. Scenarios in which you may want to have your own ingestion program include:

- You want participants to submit libraries of functions rather than executables (the ingestion program will then run them with the main function, which is the same for all participants). This allows you to use, for example, the same program to read data for everybody or to call the library functions in cross-validation loops.
- You organize active learning, query learning, time series prediction, or reinforcement learning competitions in which the code of the participants get progressively exposed to data in an interactive way.

For more information see [Building an Ingestion Program for a Competition](#).

Creating a Scoring Program

The scoring program evaluates the participants' submissions, comparing each submission with a set of reference data (solution), and then passing the resulting scores to the competition leaderboard ("Results" tab).

Each phase may have a different scoring program. For more information see [Building a Scoring Program for a Competition](#).

Creating a Starting Kit

The starting kit is a "freestyle" bundle created by the organizers, containing everything the participants need to get quickly started. A typical starting kit contains:

- A README file (use e.g. [README.md](#) or `README.ipynb` to facilitate sharing the starting kit on Github).
- A sample result submission.
- A sample code submission. The readme file contains instruction and/or sample code to read/display data, and prepare a sample submission. We provide a [template starting kit -- also downloadable as zip --](#) for the [Iris challenge](#).

Creating a Competition Bundle

The next step is creating a competition bundle. For detailed instructions, see [Building a Competition Bundle](#).

Running a Competition

Once your competition is up and running, you can manage it from your [CodaLab Dashboard](#). For more details, see [Running a Competition](#).