

# FeedBuzz

A question scorer prototype, *not totally based from BuzzFeed*.

## Contents

1. [Introduction](#)
2. [Setup](#)
3. [Usage](#)
4. [Presets](#)
5. [Scoring System](#)
6. [FeedBuzz Lingo](#)
7. [JSON Result Data](#)
8. [Visions](#)

## Introduction

In short, FeedBuzz is a console test-taking program. It takes a Feedbuzz File, and exports test scoring data. An answer can weigh how much it rewards and penalizes the overall test score.

## Setup

It is recommended to have Python 3.12 installed to run this program. Although the source code does not depend on the features from newer Python versions, the program *should* be compatible with Python 3.

Make sure the following source code files are in the same folder:

1. `main.py`
2. `reader.py`
3. `test.py`

## Usage

`feedbuzz` is `python main.py` assuming if you run it in a directory or a folder containing `main.py`

```
feedbuzz [input file] [--output output_file] [--maximum] [--ranges] [--questions] [--help]
```

- `[input file]` Take an [Feedbuzz Lingo](#) input file.
- `--output output_file` Output a file with the name `output_file`. If left unspecified, it will output a file named `results.json`.
- `--maximum` Calculate the maximum possible points in a test.
- `--ranges` List all the judgment ranges in a test.
- `--questions` List all the questions in a test.
- `--help` Show a brief help message.

## Presets

Feedbuzz bundles with a few presets, which you can find them at the presets folder.

### presets/business.q

A test examining entraprenuerial traits.

### presets/demo.q

A test showcasing the basic features of FeedBuzz.

### presets/grit.q

A test based on the Grit Scale by Angela Duckworth.

## Scoring

The scoring system is customizable [with the `Deduction` attribute](#). A score cannot be negative but instead clamped to zero.

### Punishing Scoring

Single-choice questions will lower your test score from a wrong answer. Multiple-choice questions will lower your test score if losses outweigh.

```
Question "What can be 10?":
  Choice Multiple
  Answer "8 + 2" Gain 1
  Answer "2 + 1" Loss 1 # selected
  Answer "9 + 8" Loss 1 # selected
  Answer "5 + 7" Loss 1 # selected
;

# This has a total penalty of 3
```

### Sparing Scoring

*To make a progressive score system in your test, only use penalties in multiple-choice questions.*

Single-choice questions are the same as in a test with punishing scoring unlike multiple-choice questions. If losses outweigh, multiple-choice questions have no reward and no loss.

```
Question "What can be 10?":
  Choice Multiple
  Answer "8 + 2" Gain 1
  Answer "2 + 1" Loss 1 # selected
  Answer "9 + 8" Loss 1 # selected
  Answer "5 + 7" Loss 1 # selected
;

# This has no penalty because the test uses the sparing scoring system.
```

## FeedBuzz Lingo

Feedbuzz has a descriptive language to run questions. The language is simple to learn as the syntax is self-descriptive. All input files should be written in Feedbuzz Lingo.

The typical file format of a FeedBuzz Lingo file ends with ".q" standing for questions.

### Parts

1. [Sample File](#)
2. [Section Pattern](#)
3. [Quoted Strings](#)
4. [Comments](#)
5. [File Structure](#)
6. [Test Metadata](#)
7. [Judgment Data](#)
8. [Question Entry](#)

### Sample File

```
Test:
  Name "Sample Test"
  TimeLimit 120
;

Scoring:
  At 2 "Good enough I guess."
  At 1 "Cringe"
  At 0 "No common sense you have"
;

Question "Which poison would you pick?":
  Choice Single
  Answer "Right" Gain 1
```

```

    Answer "Neutral"
    Answer "Wrong" Loss 99
;

Question "What is not the third color of the rainbow?":
    Choice Multiple
    Ordering Alphabetical
    Answer "red" Gain 1
    Answer "orange" Gain 1
    Answer "yellow" Loss 1
    Answer "green" Loss 1
;

```

## Section Pattern

This pattern is to describe a section, and below is the layout.

```

HeaderName [HeaderValues]:
    Attribute AttributeValue
    Attribute2 AttributeValue2
;

```

Start defining attributes with a colon after the data header. End a section with a semicolon.

NOTE: *Although Feedbuzz ignores whitespace, keep indentation consistent.*

```
Test: Title "One Line Sample";
```

## Quoted Strings

Strings of text can have double or single quotes in between, and the quotes must be the same. To put quotations in strings without confusing Feedbuzz, place a backtick before a quote.

```

Question "Sample":
    Answer 'Foo'
    Answer "Bar"
    Answer "The duck says, \"Quack!\""
;

```

## Comments

Like Python, begin a one-line comment with a hashtag.

```

# This is a weird question.
# Why do we have this in the test?

Question "Do you support [insert controversial policy here]?:
    Answer "Yes." Gain 99999
    Answer "No." Loss 0
    Answer "I prefer not to say." Loss 99999999
;

```

## File Structure

A file can only have one **Test** and one **Scoring** entry. You can have as many questions and judgments as you like. *However, the location of the questions define the order.* Judgment descriptions are sorted with their level, but keep it readable by manually sorting.

```

Test:
    # test data here
    At 5 "Level 5"
    At 10 "Level 10" # This should be on top of level 5
    At 0 "Level 0"
;

Scoring:
    # judgment data here
;

Question "Question1":
    # question data here
;

Question "Question2":
    # question data here
;

```

```

;

Question "QuestionX":
    # question data here
;

```

## Test Data

This is the basis of a FeedBuzz Lingo file. You can only declare it once.

```

# default values when unspecified

Test:
    Name "Test Name"
    Description "Test description"
    Deduction Sparing
    TimeLimit 0
;

```

### • Metadata

FeedBuzz displays the name and description before a test begins. [The Results JSON includes this metadata.](#)

- `Title [title]` | The title of a test.
  - `[title]` | *quoted string*
- `Description [description]` | The description of a test.
  - `[description]` | *quoted string*
- `TimeLimit [seconds]` | The time limit of a test in seconds. Set to zero to impose unlimited time.
  - `[seconds]` | *positive integer*
- `Deduction [difficulty]` | The scoring system difficulty of a test.
  - `[difficulty]` | *Sparing*

## Judgment Data

This is the end result messages of a test. Feedbuzz will sort the positions if out of order, though it is best to keep things readable by manually sorting.

```

Scoring:
    At 4 "A score at least more than 4 is displayed"
    At 2 "A score at least more than 2 is displayed"
    At 0 "A score below anything above is displayed"
;

```

- `Scoring` | **Scoring data**
  - `At [point] [description?]` | Pinpoint a message starting with `[point]`
    - `[point]` | *number* | A requirement if a score is at least the amount of it.

You can look how FeedBuzz reads judgment data by running `feedbuzz [input file] --ranges`

## Question Entry

This is the basis of a question.

```

# default values when unspecified

Question "[question name placeholder]":
    Choice Single
;

```

**Note: The question entry must have at least one answer, otherwise FeedBuzz will not accept it.**

- `Answer "[answer contents]" [effect?] [score?]` | Adds an answer to the question.
  - `[answer contents]` | *quoted string* | The contents of an answer
  - Scoring (Optional) (If left unspecified, this has no reward or penalty)
    - `[effect]` | `Gain` or `Loss` | The scoring effect of an answer
    - `[score]` | *positive integer* | The weight of the effect of an answer
- `Choice [type]` | Make the question have one or many choices selected at once.
  - `[type]` | `Single` or `Multiple`

## Samples

```
Question "What is 1+1?":
  Choice Single

  Answer "0"
  Answer "1"
  Answer "2" Gain 1
  Answer "3"
;
```

## JSON Result Data

FeedBuzz provides JSON for easily processing data into other services you wish.

- **metadata**
  - **title** | string | The title of the test taken.
  - **description** | string | The description of the test taken.
  - **time\_limit** | number | The time limit of the test taken. (If the time limit is zero, it means there is no time limit.)
- **time**
  - **started** | string | Date and time of when the test started.
  - **finished** | string | Date and time of when the test finished.
- **performance**
  - **score** | number | The total score the test taker ended up.
  - **maximum** | number | The maximum amount of points possible in the test.
  - **score\_description** | string | The description of a score range relating to the score.
  - **overdue** | boolean | If a test taker finished later beyond the time limit.
- **questions** | array | An array of questions in the test
  - - **contents** | string | The title of the question
    - **multi\_choice** | boolean | If the question is multi-choice. False if it is not multi-choice.
    - **answers** | array | An array of answers in the question
      - **contents** | string | The contents of the answer shown under the question.
      - **score** | number | The reward or penalty of the answer. Negative for penalty, positive for reward.
      - **picked** | boolean | If the test taker has selected the answer.

## Visions

This is the only beginning, and it is a prototype that gets the idea done. We have visions of shaping this to a polished web application for user convenience. Perhaps a service platform, despite being despicably similar to a \_\_\_\_\_ and \_\_\_\_\_ I don't think the laymen would use this if this console application released today.

For now, this is a passion project made for a family member and for a school project.