Experimental design and statistical analysis with IbexFarm and RStudio

Experimental Syntax

19th Novermber 2025

The scientific method

Observation/Question

↓

Research

↓

Hypothesis

↓

Experiment

↓

Data collection

↓

Analysis

↓

Conclusions

↓

Observation/Question

↓

etc.

Research topic

↓

Literature review

↓

Predictions

↓

Hypothesis testing

↓

Experiment running

↓

Data examination

↓

Comparing data with hypothesis

↓

Circling back with new observations

↓

etc.

Aims of an experiment

1. Make an observation/Ask a question
2. Formulate hypotheses
3. **Operationalise predictions**
4. **Experimentally test the predictions**

Rules for setting up an experiment (cf. B. Hemforth's lecture)

Rule 1 : Formulate the research question in detail and using the terms of the chosen experimental approach.

Rule 2 : Achieve minimal contrasts (vary only what you want to vary).

Rule 3 : Reduce "noise", control variability outside the experiment.

Rule 4 : Adapt the number of items / subjects to the expected effects.

Operationalising a hypothesis : Example

You want to test whether adjectival agreement in coordination for French follows masculine resolution (1) or closest conjunct agreement (2)

(1) Des étudiants et étudiantes **nouveaux**.
The student.M.PL and students.F.PL new.M.PL
De **nouveaux** étudiantes et étudiants
The new.M.PL student.F.PL and students.M.PL

(2) Des étudiants et étudiantes **nouvelles**.
The student.M.PL and students.F.PL new.F.PL
De **nouvelles** étudiantes et étudiants
The new.F.PL student.F.PL and students.M.PL

$\Rightarrow$ You want to know what strategy people will use, and whether adjectival position (pre- or post-nominal) will have an effect

Operationalising a hypothesis : Example

You formulate a **hypothesis** : with *et* ('and') coordination, the adjective will agree with the closest conjunct.

But, you also suppose that the position of the adjective matters.

You narrow down your hypothesis : for prenominal adjectives, masculine resolution is preferred, while for postnominal adjectives, closest conjunct agreement is preferred

From this hypothesis, you get the following **predictions** :

- Prenominal adjectives : CCA <MascRes
  Afem Nfem *et* Nmasc <MASC RES **Amasc** Nfem *et* Nmasc

- Postnominal adjectives : CCA >MascRes
  CCA Nmasc *et* Nfem **Afem** >MASC RES Nmasc *et* Nfem **Amasc**

From hypothesis to experiment

We want to know if one resolution strategy is preferred to another

$\rightarrow$ for corpus data, preferences emerge through occurrence counts

$\rightarrow$ for experimental data, one way to probe preferences is to compare sentence **acceptability ratings**

We consider 2 factors : adjective position and agreement strategy (our variables)
$\rightarrow$ 4 possibilities (conditions)

Variable types

Dependant variables (DV) : what you measure
Acceptability Judgment Tasks measure explicit responses
(Likert scale ratings on a 1-10 scale)

Independant variables (IV) : what you are testing for
The factors you are varying to operationalise your RQ
(in our case, conjunct order & agreement strategy)
There are multiple types of variables that you can vary :

- **Invoked variables :** invoked from participants using questionnaires (age, gender, language(s) spoken, etc.)
- **Provoked variables :** extracted from the physical or social environment of the subjects (any experimental variation)
- **Factors :** at least two levels of the IV that you will compare (CCA/MascRes, men/women, etc.)

Covariates : controlled variables

Confounds : variables that are not controlled for/tested directly and which may impact observations
(e.g. participant concentration, possible reading impairments, etc.)

Design types

Factorial designs : 2 or more independent variables
(across factors : each combination of factors is a condition)

- **Within-Subject** : 1 participant sees every condition of each item

- **Between-Subject** : 1 participant sees one (random) condition per item

⇒ Between-Subject designs are preferable
(but require more participants to reach the same level of statistical power)

Statistical power : the chance of finding an effect, if there is one
Conditions : different levels of the independent variable

- **Within-Item** : 1 item set contains all conditions

- **Between-Item** : 1 item exists in 1 condition only

⇒ Within-Item designs are preferable : allows us to control item variability (fewer confounding factors) and give greater statistical power

Best case : Within-Item and Between-Subject design

Latin Square designs (Euler, 1773)

Create n distinct experimental lists for an n-condition experiment such that each list has exactly one condition from each item set

Example of a 2x2 Latin Square design (2 factors, 2 levels each) :

Items	**List 1**	**List 2**
Item 1	Condition 1 Condition 2	Condition 2 Condition 1
Item 2	Condition 2 Condition 1	Condition 1 Condition 2

Note for later : IbexFarm will automatically divide your experimental items into Latin square designs

Things to keep in mind

Stimuli should be **counterbalanced** : each participant should see an identical number of items, each item should only appear in one condition

Stimuli should be **randomised** : no two participants should see the items in the same order

Ibexfarm

Ibex was created by Alex Drummond, it is a linguist-designed internet-based experimental interface for conducting a wide variety of experiments

$\Rightarrow$ what you will use here is a locally-hosted clone of IbexFarm (LLF servers are GDPR compliant)

Experiments are sent out to participants via a URL link

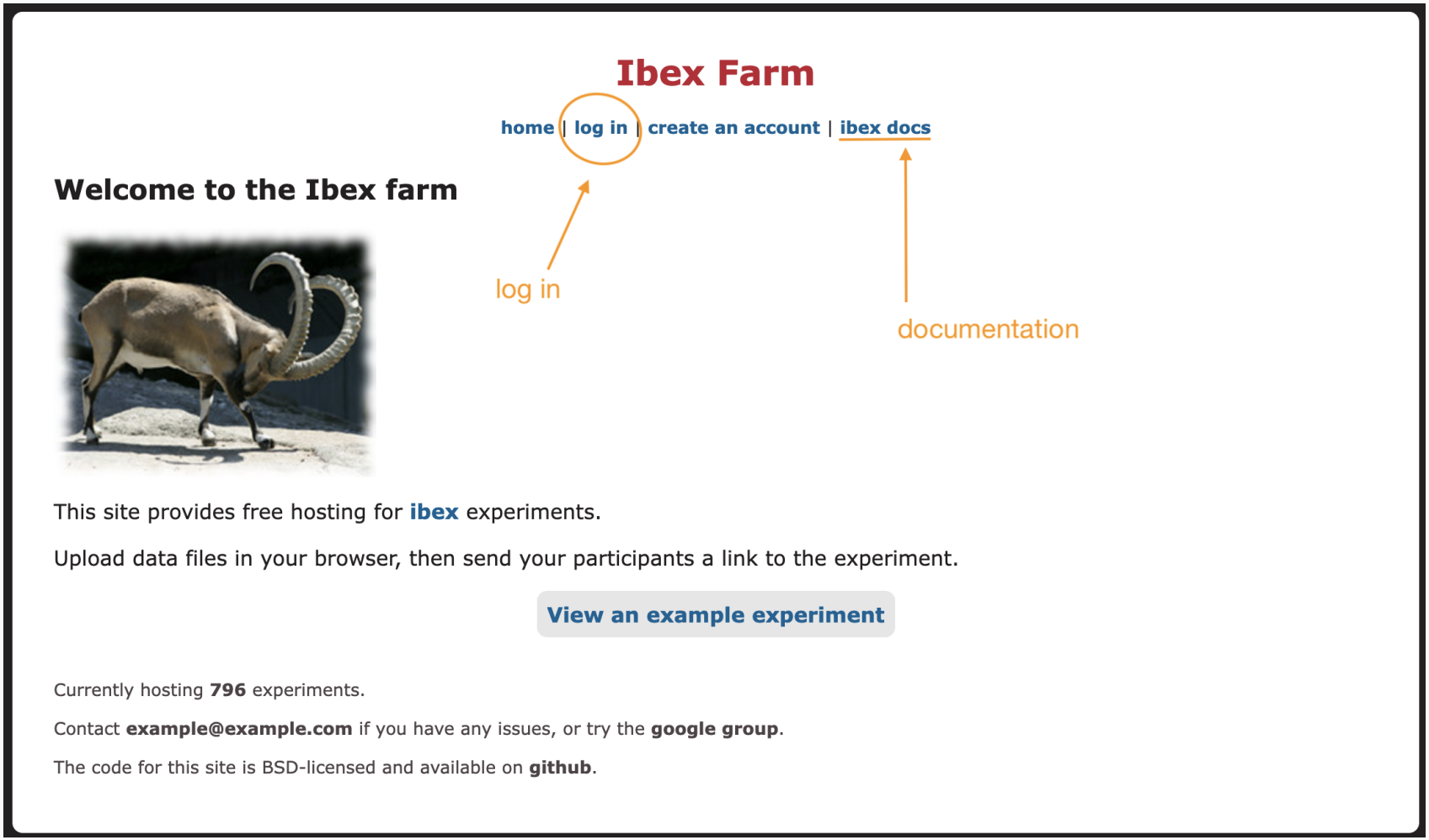
IbexFarm mainly uses JavaScript, but there is a wide set of ready made "controllers" (e.g. self-paced reading, acceptability judgments...)
$\Rightarrow$ basic knowledge is enough !

Each controller has different properties or arguments : they are shared by all instances of a given controller in your experiment
$\Rightarrow$ these controllers can be customised for more complex designs

Navigating Ibex

Step 1 : Load the Ibex website

Link : https://ibex.llf-paris.fr/



Navigating Ibex

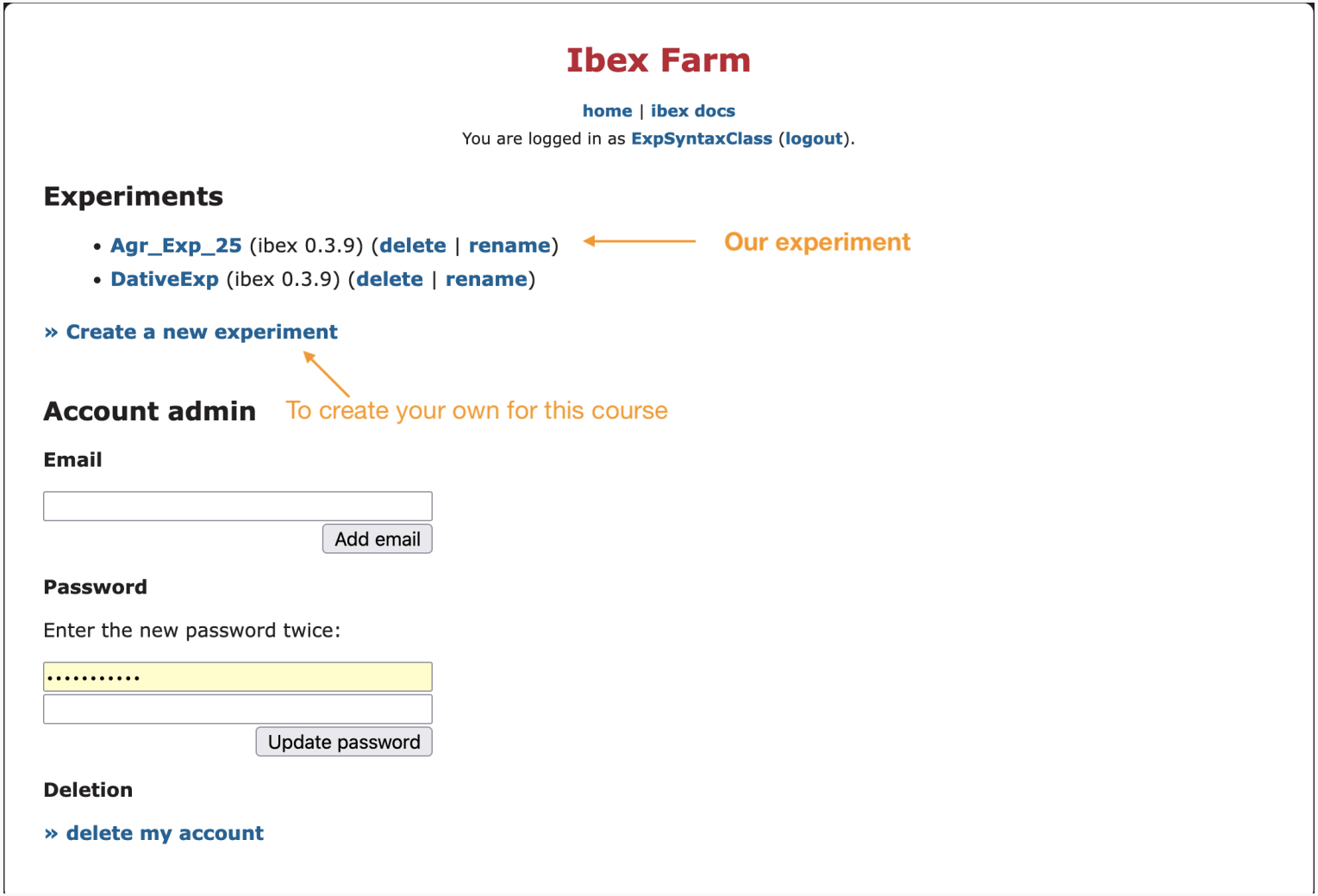
Step 2 : Log in to the class account

Username : ExpSyntaxClass

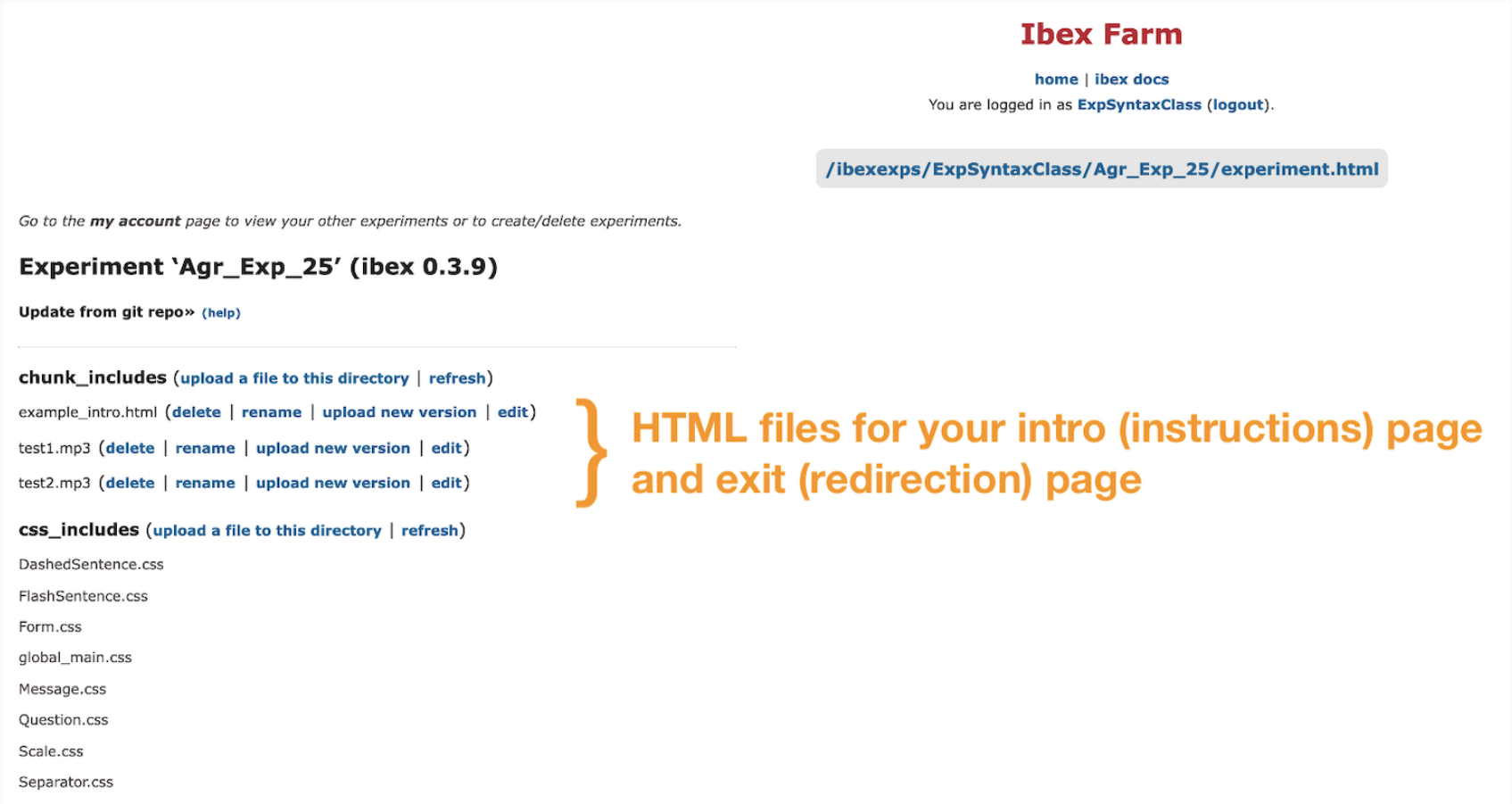
Password : ExpSyntaxClass

Step 3 : Select (or create) your experiment

Experiment : Agr_Exp_25



Step 4 : Code your experiment



Navigating Ibex

Step 4 : Code your experiment

Coding your experiment in Ibex : Instructions page

Ibex's default instructions page

This experiment has two parts. In the first part, you will be required to read a number of English sentences one word at a time. You can advance to the next word by pressing the space bar.

In the second part, you will be asked to "rate" some sentences on a scale of 1-7 (with 1 bad and 7 good).

[Privacy statement, etc., etc.]

(The form below is just a rough example showing all of the input controls that work properly with Ibex. Obviously it would not necessarily be appropriate to collect all of this data on subjects.)

Name:

Age:

Sex: Male Female

I am doing this experiment for credit

I consent

Write something:

Instructions page

Instructions

Start by explaining what the task will entail
→ **do not** give away your research question
(don't specify what you are testing for)

Then describe what the task will look like
(what acceptability is, how to rate sentences, attention checks, etc.)
→ explain that there are no good or bad answers
→ be precise (participants are unpredictable creatures)

Privacy statement

cf. Ethics Principles of free and informed consent, and data retraction rights

Meta-data questionnaire

You may ask questions about participants that may be relevant to your research question (they should be justified and should not allow you to identify your participants)

Coding your experiment in Ibex : Shuffle sequence

Shuffle sequence

```
var shuffleSequence = seq("intro","practice", "sep","setcounter", //set the sequence in order with Intro (instructions page) -> practice items -> separator -> set the participant counter
                      rshuffle(//first shuffle between target/filler/control items
                               rshuffle("before_AfemNfem","before_AmascNfem","after_AmascNfem","after_AmascNmasc"), //embed shuffle to shuffle target item conditions
                               rshuffle("-hum+dist","+hum+dist", "-hum-dist", "+hum-dist"),//embed shuffle to shuffle filler item conditions
                               rshuffle("gramm", "ungramm")),  //embed shuffle to shuffle control item conditions
                       "send_results", "outro"); //sequence again to send results to the server -> exit page

// watch out for punctuation/syntax errors with your shuffle sequence!
```

Shuffle sequence

```
var shuffleSequence =
    seq("intro","practice", "sep", "setcounter"),
    rshuffle(
        rshuffle("cond_a","cond_b","cond_c","cond_d"),
        rshuffle("filler_a","filler_b"),
        rshuffle("control_a", "control_b")),
    "send_results", "outro");
```

1. Set the sequence order with :
   Intro (instructions page) → practice items → separator → set the participant counter
2. First shuffle between target/filler/control items
3. Embed shuffle to shuffle target item conditions
4. Embed shuffle to shuffle filler item conditions
5. Embed shuffle to shuffle control item conditions
6. Sequence again to send results to the server → exit page

Controllers and item types

```
var practiceItemTypes = ["practice"]; // these are your practice items which you will define below

var defaults = [ //these defaults come with Ibex, I've deleted the ones that are not relevant for our introduction (but you may find them in the example_data.js file)
    "Separator", {
        transfer: 2000,// specify how long the separator page is displayed
         normalMessage: "The practice session is over. The experiment will begin shortly." }, // here is where you specify what the separator says


    "AcceptabilityJudgment", {
        as: ["1", "2", "3", "4", "5","6","7","8","9","10"], // you may customise the scale if you do not want a 1-10 task
        presentAsScale: true,
        instructions: "Click on the number (or box) which corresponds to your judgment.", // you may customise the instructions that go with your scale
        leftComment: "(unnatural)", rightComment: "(completely natural)",
        hideProgressBar: false,
    },

    "Question", {
        hasCorrect: true,
        hideProgresssBar : false, //these are for your comprehension questions
     // instructions: " "      // you may specify instructions with this syntax

    },


    "Form", {
        hideProgressBar: true,
        continueOnReturn: true,
        saveReactionTime: true,
        continueMessage: "Click here to continue", //these are the instructions for the form element (your intro page)
        countsForProgressBar: true
    }

]; //don't forget this before you add your items variables, otherwise you get an error message
```

Controllers and item types

Start by defining you practice item as a variable :

```
1 var practiceItemTypes = ["practice"];
```

For Acceptability judgment tasks, you use the following controllers :

- **Separator** : specify the separator page (between the practice items and the beginning of the task) message and display time
- **AcceptabilityJudgment** : the controller for the experiment type, you can customise the scale and instructions
- **Question** : the controller for the comprehension questions, you can customise the instructions and a pre-coded answer (`hasCorrect: true`)
- **Form** : the html elements in the `chunk_includes` directory (instruction and exit pages)

Sequence elements : item variables

```
var items = [


    ["sep", "Separator", { }], // this is a small page between the practice item and the actual beginning of the task

    ["setcounter", "__SetCounter__", { }], //this sets the participant counter to the current participant

    ["intro", "Form", {html: { include: "intro.html" }} ], //includes your introduction html file as a variable in the experiment sequence
    ["outro", "Form", {html: { include: "outro.html" }} ], //same for the exit page

    ["send_results", "__SendResults__", {}], //send results to the server
```

Sequence elements : item variables

Start the variable definition :

```
var items = [
```

Then add :
The separator

```
["sep", "Separator", { }],
```

The participant counter

```
 ["setcounter", "__SetCounter__", { }],
```

The intro and outro html pages

```
["intro", "Form", {html: { include: "intro.html" }} ],
["outro", "Form", {html: { include: "outro.html" }} ],
```

The send_results command

```
["send_results", "__SendResults__", {}],
```

Example : separator and item

progress

The practice session is over. The experiment will begin shortly.

progress

Des ananas ou cerises délicieuses sont disponibles au marché.

(unnatural) 1 2 3 4 5 6 7 8 9 10 *(completely natural)*

Click on the number (or box) which corresponds to your judgment.

Item syntax

Each item condition is enclosed in square brackets, and ends with a comma

```
1 [["cond1",101],"AcceptabilityJudgment",
2     {s:"Les étudiantes et étudiants nouveaux"},
3     Question, {q:"Les étudiants sont-ils anciens ?",
4         as: ["Oui.", "Non."], hasCorrect :"Non."}],
```

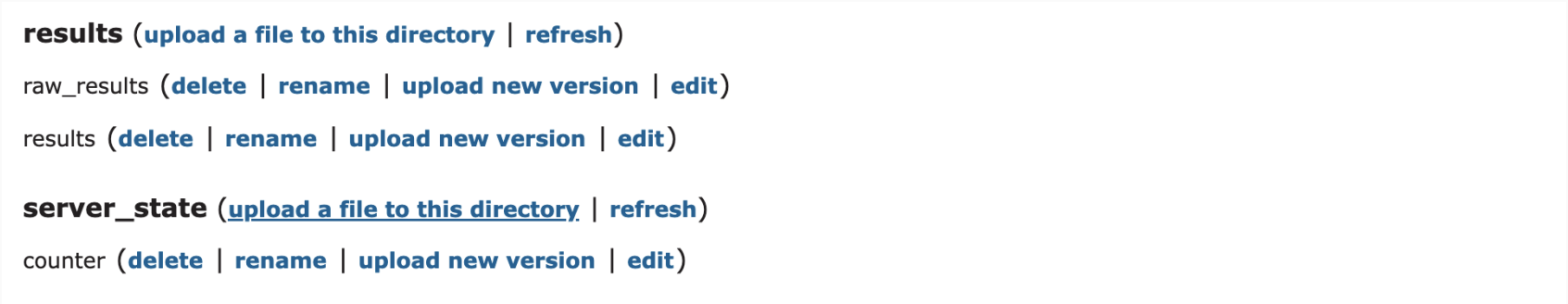
- `["cond1",101],` : specify the condition between double quotes, then the item number without quotes
- `"AcceptabilityJudgment",` : specify the controller
- `{s:"Les étudiantes et étudiants nouveaux"},` : the item text (of type `s` for sentence)
- `Question,` : specify the controller for questions
- `{q:"Les étudiants sont-ils anciens ?",` : the question text (of type `q` for question)
- `as: ["Oui.", "Non."],` : the two possible choices for the answer
- `hasCorrect :"Non."}],` : the expected answer

Coding your experiment in Ibex : Items

Watch out for :

- **Item numbers** : they should not overlap between practice/fillers/controls
- **Syntax** : Ibex is **very** finicky with syntax, especially brackets and punctuation (double/triple check them !)
  $\rightarrow$ 9/10 errors are because of a misplaced comma or bracket
- **Server crashing** : this happens more than you think, so code your items and instruction pages locally (text editor) and copy them into Ibex once they're ready

Getting results from Ibex



Use the **results** file (it is comma separated, but you can download it as a txt file)
$\rightarrow$ the raw_results file is the same but much harder to navigate

Reading results from Ibex

```
#     Col. 7: Group.
#     Col. 8: Question (NULL if none).
#     Col. 9: Answer.
#     Col. 10: Whether or not answer was correct (NULL if N/A).
#     Col. 11: Time taken to answer.
# Line 2:
#     Col. 1: Time results were received.
#     Col. 2: MD5 hash that hopefully uniquely identifies participant.
#     Col. 3: Controller name.
#     Col. 4: Item number.
#     Col. 5: Element number.
#     Col. 6: Type.
#     Col. 7: Group.
#     Col. 8: Sentence (or sentence MD5).
1727113687,c29be33e69baa4c36d171e1c6245252d,Question,6,1,practice,102,Le directeur va partir Ã  la retraite.,Non,1,9079
1727113687,c29be33e69baa4c36d171e1c6245252d,AcceptabilityJudgment,7,0,practice,103,Avant le dÃ©veloppement de Rome%2C l'Italie Ã©tait composÃ©e de plusieurs cultures et civilisations%2C desquelles on ne sait pas
encore tout.
1727113687,c29be33e69baa4c36d171e1c6245252d,AcceptabilityJudgment,7,0,practice,103,NULL,8,NULL,15603
1727113687,c29be33e69baa4c36d171e1c6245252d,AcceptabilityJudgment,291,0,fd,71,Marc est une pharmacienne de quartier.
# Columns below this comment are as follows:
# 1. Time results were received.
# 2. MD5 hash that hopefully uniquely identifies participant.
# 3. Controller name.
# 4. Item number.
# 5. Element number.
# 6. Type.
# 7. Group.
# 8. Question (NULL if none).
# 9. Answer.
# 10. Whether or not answer was correct (NULL if N/A).
# 11. Time taken to answer.
1727113687,c29be33e69baa4c36d171e1c6245252d,AcceptabilityJudgment,291,0,fd,71,NULL,8,NULL,11279
1727113687,c29be33e69baa4c36d171e1c6245252d,Question,291,1,fd,71,Est-ce que la phrase parle de Lola ?,Non,1,4335
# The lines below this comment are in groups of 2.
# The formats of the lines in each of these groups are as follows:
#
# Line 1:
#     Col. 1: Time results were received.
#     Col. 2: MD5 hash that hopefully uniquely identifies participant.
#     Col. 3: Controller name.
#     Col. 4: Item number.
#     Col. 5: Element number.
#     Col. 6: Type.
#     Col. 7: Group.
#     Col. 8: Sentence (or sentence MD5).
# Line 2:
#     Col. 1: Time results were received.
#     Col. 2: MD5 hash that hopefully uniquely identifies participant.
#     Col. 3: Controller name.
#     Col. 4: Item number.
#     Col. 5: Element number.
#     Col. 6: Type.
#     Col. 7: Group.
#     Col. 8: Question (NULL if none).
#     Col. 9: Answer.
#     Col. 10: Whether or not answer was correct (NULL if N/A).
#     Col. 11: Time taken to answer.
```

Lines that start with # are comments that include metadata
(and tell us which variables each comma-separated values correspond to)

Reading results from Ibex

The variables returned by Ibex are :

- **time** (unix time stamp)
- **participant ID** (automatically generated random unique identifier)
- **controller** used for the experiment
- **item number**
- **element number** (ignore)
- **type** (practice, filler, target)
- **group** (encodes randomisation choices)

Result interpretation (cf. B. Hemforth's lecture)

If we reject the null hypothesis with α=.05 :

There is a low (p <0.05) probability that the sample of participants and items we chose is not representative of the population.

$\Rightarrow$ High chance to replicate the results in a new study

$\Rightarrow$ Typical interpretation : There is a high probability that there exists a real difference in the population

If we fail to reject the null hypothesis :

There is no difference in the population

OR

We were not able to detect the difference in the population

Important to know

Type I errors : we rejected the null hypothesis when in reality it was true

Type II errors : we failed to reject the null hypothesis when in reality it was false

Statistical power for experiments : the greater the sample size (amount of observations per condition = amount of participants), the smaller the variance

Adding random effects : check sources of the variance in the DV

- **Systematic** variance : variability related to the manipulation of IVs and other identified variables.
- **Random** Variance : Variability whose source is unknown.

Inferential stats : Is there systematic variance beyond random variability ? (Is there a signal in the noise ?)

Ethics Principles (cf. B. Hemforth's lecture)

Scientific integrity

- Scientific protocols that involve humans must be valid
- Researchers must be qualified for the proposed research
- Conflicts of interest must be avoided

Integrity of the individual

- Potential benefits of the research must outweigh risks and drawbacks
- Risks must be clearly identified and explained to participants, even if they are rare or unlikely.
  All types of risks must be considered (not only physical, but psychological, moral, economic, social)

Ethics Principles (cf. B. Hemforth's lecture)

Respect for dignity and privacy

- Researchers must respect the dignity and privacy of participants
- Researchers must ensure the confidentiality of the nominative information collected

Personal Autonomy

- The decision whether or not to participate in a research project must be free and informed
- Free : without pressure from researchers or any other person, especially in cases where people are vulnerable and/or captive (students, employees, people receiving services, etc.).
  The decision to participate or not should not have an impact on the services received
- Informed : Potential participants must make their decision based on complete and clear information about the research.
- For the participation of minors, parental consent is required