1. Generate a new app:

* Write on the command window:

**otree startapp feedback**

1. Add the new app as an entry to the settings.py

* Copy one of the existing ones and modify the information such that feedback is the app used. Feel free to adjust the names. At least one demo participant needed.

1. Modify the models.py to create the following variables for the Player:

languages = models.IntegerField(  
 label="I recognize the different coding languages (HTML5, JavaScript, Python, Django)"  
 " and know when to use which",  
 choices=[  
 [1, 'Unsure'],  
 [2, 'Agree partially'],  
 [3, 'Agree']  
 ],  
 widget=widgets.RadioSelect,)  
dynamic = models.IntegerField(  
 label="I can dynamically change the context of a web page",  
 choices=[  
 [1, 'Unsure'],  
 [2, 'Agree partially'],  
 [3, 'Agree'],  
 ],  
 widget=widgets.RadioSelect,)  
syntax = models.IntegerField(  
 label="I know the basic syntax of each coding language (HTML5, JavaScript, Python, Django)"  
 " and how to search advice online",  
 choices=[  
 [1, 'Unsure'],  
 [2, 'Agree partially'],  
 [3, 'Agree'],  
 ],  
 widget=widgets.RadioSelect,)  
structure = models.IntegerField(  
 label="I understand the basic purpose of the main oTree files and how they come together",  
 choices=[  
 [1, 'Unsure'],  
 [2, 'Agree partially'],  
 [3, 'Agree'],  
 ],  
 widget=widgets.RadioSelect,)  
debugging = models.IntegerField(  
 label='I can decode error messages (command output and oTree error messages) and find potential solutions',  
 choices=[  
 [1, 'Unsure'],  
 [2, 'Agree partially'],  
 [3, 'Agree'],  
 ],  
 widget=widgets.RadioSelect,)  
inspect = models.IntegerField(  
 label='I can inspect the source code on the browser',  
 choices=[  
 [1, 'Unsure'],  
 [2, 'Agree partially'],  
 [3, 'Agree'],  
 ],  
 widget=widgets.RadioSelect,)  
difficulty = models.IntegerField(  
 label='I found tasks',  
 choices=[  
 [1, 'Too easy'],  
 [2, 'Ok, varied, both easy and difficult'],  
 [3, 'Too difficult (but I understood what was being asked)'],  
 [4, 'Buggy, with many mistakes, too long, or unclear']  
 ],  
 widget=widgets.RadioSelect,)  
stuck = models.StringField(  
 label='If you can remember an occasion where you got stuck, please describe here, when it happened.',  
 null=True,  
 blank=True,  
)  
useful = models.IntegerField(  
 label='I found most useful (choose only one)',  
 choices=[  
 [1, "Learning basic languages: html, Django, JavaScript, Python"],  
 [2, "Learning about debugging (console log, code inspection, oTree error feedback)"],  
 [3, "Learning about the basic structure of oTree and how to modify it"],  
 [4, "Learning about the online resources and how to use them"],  
 ],  
 widget=widgets.RadioSelect,)  
success = models.LongStringField(  
 label='What did you succeed in today?',  
 null=True,  
 blank=True,  
)  
future = models.LongStringField(  
 label='What do you want to learn in the next session?',  
 null=True,  
 blank=True,  
)  
comment = models.StringField(  
 label='You can leave a free comment here:',  
 null=True,  
 blank=True,  
)

1. Modify the pages.py by adding the variables to class MyPage:

form\_model = 'player'  
form\_fields = [  
 'languages',  
 'dynamic',  
 'syntax',  
 'structure',  
 'debugging',  
 'inspect',  
 'difficulty',  
 'stuck',  
 'useful',  
 'success',  
 'future',  
 'comment',  
]

1. Add the formfields to the MyPage.html with the appropriate Django command
2. Answer the feedback questionnaire
3. Then go to the otree monitor pages, download your data as an excel file and send it by email to Essi.

Goals left for the second day:

In the next session, besides continuing with the previous learning goals, we will also work on how to break down an experimental design into small and tangible coding problems, and how to follow ‘good coding practices’, such as, avoiding repetition,….