// Remove command prefix  
PennController.ResetPrefix(null);  
  
// Turn off debugger - keep on when building exp., turn off when running exp.  
DebugOff();  
  
// Increment the counter for list balancing;  
// Use when sending experiment link online, unnecessary otherwise  
SetCounter("counter", "inc", 1);  
  
// Sequence of the experiment  
Sequence("consent", "demographics", "instructions", "practice", "practice-end",  
         shuffle(randomize("experimental-trial")),  
         "done", "debriefing", "send", "completion\_screen");  
  
// Consent form  
newTrial("consent",  
    newHtml("consent\_form", "consent.html")  
        .cssContainer({"width":"1000px"})  
        .checkboxWarning("You must consent in order to participate in the experiment")  
        .center()  
        .print()  
    ,  
    newButton("continue", "Continue")  
        .center()  
        .print()  
        .wait(getHtml("consent\_form").test.complete()  
                  .failure(getHtml("consent\_form").warn()))  
);  
  
// Demographics  
newTrial("demographics",  
    newHtml("demographics\_form", "demographics.html")  
        .cssContainer({"width":"1000px"})  
        .center()  
        .print()  
        .log()  
    ,  
    newTimer(20)  
        .start()  
        .wait()  
    ,  
    newFunction( ()=>window.scrollTo(0,0) )  
        .call()  
    ,  
    newButton("continue", "Continue")  
        .center()  
        .print()  
        .wait(getHtml("demographics\_form").test.complete()  
            .failure(getHtml("demographics\_form").warn()))  
);  
  
// Instructions  
newTrial("instructions",  
    newHtml("instructions", "instructions.html")  
        .cssContainer({"width":"1000px"})  
        .center()  
        .print()  
    ,  
    newTimer(20).start().wait()  
    ,  
    newFunction( ()=>window.scrollTo(0,0) ).call()  
    ,  
    newButton("continue", "Continue")  
        .center()  
        .print()  
        .wait(getHtml("instructions").test.complete())  
);  
  
// Practice  
Template("demo\_practice.csv", row =>  
    newTrial("practice",  
        // Show context  
        newText("context", row.context)  
            .center() // Center the context text  
            .print()  
        ,  
        newKey(" ")  
            .wait()  
        ,  
        getText("context")  
            .remove()  
        ,  
        newTimer(400).start().wait()  // 400ms break after context  
        ,  
        // RSVP presentation for the sentence  
        ...row.sentence.split(" ").map((word, i) => [  
            newText(`word${i}`, word)  
                .center()  
                .print()  
            ,  
            newTimer(300).start().wait(),  
            getText(`word${i}`).remove(),  
            newTimer(100).start().wait()  // 100ms break between words  
        ]).flat()  
        ,  
        // Question with two possible answers  
        newText(  "question" , row.question )  
            .center() // Center the context text  
            .print()  
        ,  
        newCanvas("answers", 500, 100)  
            .add(100, 0, newText("F", "F: " + row.SECOND))  
            .add(300, 0, newText("J", "J: " + row.FIRST))  
            .center()  
            .print()  
        ,  
        newKey("answer", "FJ")  
            .wait()  
            .log()  
        ,  
        getText("question")  
            .remove()  
        ,  
        getCanvas("answers")  
            .remove()  
    )  
    .log("group", row.group)  
    .log("context", row.context)  
    .log("sentence", row.sentence)  
    .log("question", row.question)  
);  
  
// Practice-end  
newTrial("practice-end",  
    newText("practice-end", ".הגענו לסוף האימון. הניסוי מתחיל עכשיו. אנא ודאו שהמקלדת מכוונת לאנגלית")      
        .cssContainer({"font-size": "15px", "margin-top":"2em"})  
        .center()  
        .print()  
     
    ,  
    newButton("continue", "Continue")  
        .center()  
        .print()  
        .cssContainer({"margin-top":"2em"})  
        .wait()  
);  
  
// Experimental trial  
Template("demo\_items.csv", row =>  
    newTrial("experimental-trial",  
        // Show context  
        newText("context", row.context)  
            .center() // Center the context text  
            .print()  
        ,  
        newKey(" ")  
            .wait()  
        ,  
        getText("context")  
            .remove()  
        ,  
        newTimer(400).start().wait()  // 400ms break after context  
        ,  
        // RSVP presentation for the sentence  
        ...row.sentence.split(" ").map((word, i) => [  
            newText(`word${i}`, word)  
                .center()  
                .print()  
            ,  
            newTimer(300).start().wait(),  
            getText(`word${i}`).remove(),  
            newTimer(100).start().wait()  // 100ms break between words  
        ]).flat()  
        ,  
        // Question with two possible answers  
        newText("question", row.question)  
            .center() // Center the context text  
            .print()  
        ,  
        newCanvas("answers", 500, 100)  
            .add(100, 0, newText("F", "F: " + row.SECOND))  
            .add(300, 0, newText("J", "J: " + row.FIRST))  
            .center()  
            .print()  
        ,  
        newKey("answer", "FJ")  
            .wait()  
            .log()  
        ,  
        getText("question")  
            .remove()  
        ,  
        getCanvas("answers")  
            .remove()  
    )  
    .log("group", row.group)  
    .log("set", row.set)  
    .log("condition", row.condition)  
    .log("context-type", row["context-type"])  
    .log("context", row.context)  
    .log("sentence", row.sentence)  
    .log("question", row.question)  
);  
  
// Send results manually  
SendResults("send");  
  
// Done  
newTrial("done",  
    newHtml("done", "done.html")  
        .cssContainer({"width":"720px"})  
        .center()  
        .log()  
        .print()  
    ,  
    newButton("continue", "המשך")  
        .center()  
        .print()  
        .wait()  
);  
  
// Debriefing  
newTrial("debriefing",  
    newHtml("debriefing\_form", "debriefing.html")  
        .cssContainer({"width":"720px"})  
        .checkboxWarning("יש למלא את הפרטים על מנת שנוכל להעביר את התשלום")  
        .center()  
        .log()  
        .print()  
    ,  
    newButton("continue", "לחצו להמשך")  
        .center()  
        .print()  
        .wait(getHtml("debriefing\_form").test.complete()  
                  .failure(getHtml("debriefing\_form").warn()))  
);  
  
// Completion screen  
newTrial("completion\_screen",  
    newText("thanks", "תודה על השתתפותך")  
        .center()  
        .print()  
    ,  
    newButton("end experiment", "סיום")  
        .center()  
        .print()  
        .wait()