```json
{
  "metadata": {
    "kernelspec": {
      "name": "python",
      "display_name": "Python (Pyodide)",
      "language": "python"
    },
    "language_info": {
      "codemirror_mode": {
        "name": "python",
        "version": 3
      },
      "file_extension": ".py",
      "mimetype": "text/x-python",
      "name": "python",
      "nbconvert_exporter": "python",
      "pygments_lexer": "ipython3",
      "version": "3.8"
    }
  },
  "nbformat_minor": 4,
  "nbformat": 4,
  "cells": [
    {
      "cell_type": "markdown",
      "source": "# Chapter 2 - How to write your first program",
      "metadata": {}
    },
    {
      "cell_type": "markdown",
      "source": "## 2.1 Student Registration\nCreate a program that allows a student to complete a registration form and displays a completion message that includes the user's full name and a temporary password.\n\n### Console:\n'''powershell\nRegistration Form\n\nFirst Name: Eric\nLast Name: Idle\nBirth Year: 1934\n\nWelcome Eric Idle!\nYour registration is complete!\nYour temporary password is: Eric*1934\n'''\n\n### Specifications:\n- The user's full name consists of the user's first name, a space, and the user's last name.\n- The temporary password consists of the user's first name, an asterisk (*), and the user's birth year.\n- Assume the user will enter valid data.\n",
      "metadata": {}
    },
    {
      "cell_type": "code",
      "source": "print (\"Registration Form\\n\")\n\nfirst_name = input(\"First Name: \")\nlast_name = input(\"Last Name: \")\nyear: int = input(\"Birth Year: \")\n\nprint (\"\")\nprint (\"Welcome \" + first_name + \" \" + last_name + \"!\")\n\nprint(\"Your registration is complete!\")\n\nprint(\"Your temporary password is: \" + first_name + \"*\" + year)\n",
      "metadata": {},
      "outputs": [],
      "execution_count": 1
    },
    {
      "cell_type": "markdown",
      "source": "## 2.2 - Pay Check Calculator\nCreate a program that calculates a user's weekly gross and take-home pay.\n\n### Console\n'''powershell\nPay Check Calculator\n\nHours Worked: 35\nHourly Pay Rate: 14.50\n\nGross Pay: 507.5\nTax Rate: 18%\nTax Amount: 91.35\nTake Home Pay: 416.15\n'''\n\n### Specifications:\n- The formula for calculating gross pay is:\n'gross pay = hours worked
```

* hourly rate'\n- The formula for calculating tax amount is:\n'tax amount = gross pay * (tax rate / 100)'\n- The formula for calculating take home pay is:\n'take home pay = gross pay – tax amount'\n- The tax rate should be 18%, but the program should store the tax rate in a variable so that you can easily change the tax rate later, just by changing the value that's stored in the variable.\n- The program should accept decimal entries like 35.5 and 14.25.\n- Assume the user will enter valid data.\n- The program should round the results to a maximum of two decimal places.\n",
    "metadata": {}
  },
  {
    "cell_type": "code",
    "source": "print(\"Pay Check Calculator \\n\")\n\nprint (\"\")\nworked: float = input(\"Hours Worked: \")\n\npay_rate: float = input(\"Hourly Pay Rate: \\n\")\n\ngross_pay = float(worked) * float(pay_rate)\n\nTAXRATE: float = 18.0\n\ntax_amount = gross_pay * (TAXRATE / 100)\n\ntake_home_pay = gross_pay - tax_amount\n\n\n#prints gross pay taxrate tax amount and take home pay\nprint(\"Gross Pay: \" + ('%.2f'%gross_pay))\nprint(\"Tax Rate: \" + ('%.2f'%TAXRATE))\nprint(\"Tax Amount: \" + ('%.2f'%tax_amount))\nprint(\"Take Home Pay: \" + ('%.2f'%take_home_pay))\n",
    "metadata": {},
    "outputs": [],
    "execution_count": 2
  },
  {
    "cell_type": "markdown",
    "source": "## 2.3 - Travel Time Calculator\nCreate a program that calculates the estimated hours and minutes for a trip.\n\n### Console\n'''powershell\nTravel Time Calculator\n\nEnter Miles: 200\nEnter Miles per Hour: 65\n\nEstimated Travel Time\nHours: 3\nMinutes: 5\n'''\n\n### Specifications\n- The program should only accept integer entries like 200 and 65.\n- Assume that the user will enter valid data.\n\n### Hint\n- Use integers with the integer division and modulus operators to get hours and minutes.",
    "metadata": {}
  },
  {
    "cell_type": "code",
    "source": "print(\"Travel Time Calculator \\n\")\n\nmiles: float = input(\"Enter Miles: \") #200\n\nmiles_per_hour: float = input(\"Enter Miles per Hour: \") #65\n\n#get the amount of hours driven\n\nhours_driven = float(miles) / float(miles_per_hour)\n\nprint(hours_driven)\n\n\n#rounds to get int\nhours_int = round(hours_driven, 0)\n\nprint(hours_int)\n\n#gains the decimal\nminutes_decimal = hours_driven - hours_int \n\nprint(minutes_decimal)\n\n#times it by the amount of minutes in an hour\nminutes_left = minutes_decimal * 60\n\nprint(minutes_left)\n\n#rounds minutes_left\nminutes_int = round(minutes_left)\n\nprint(minutes_left)\n\nprint(\"\\nEstimated Travel Time\")\nprint(\"Hours: \" + str(hours_int))\nprint(\"Minutes: \" + str(minutes_int))\n",
    "metadata": {},
    "outputs": [],
    "execution_count": 3
  }
 ]
}