```
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": [],
      "collapsed_sections": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "markdown",
      "source": [
        "# Basic Python"
      ],
      "metadata": {
        "id": "McSxJAwcOdZ1"
      }
    },
    {
      "cell_type": "markdown",
      "source": [
  "## 1. Split this string"
 ],
 "metadata": {
  "id": "CU48hgo4Owz5"
 }
},
{
 "cell_type": "code",
 "source": [
  "s = \"Hi there Sam!\""
 ],
 "metadata": {
  "id": "s07c7JK7Oqt-"
 },
 "execution_count": 2,
 "outputs": []
},
{
 "cell_type": "code",
 "source": [
  "x=s.split()\n",
  "print(x)\n"
 ],
 "metadata": {
  "id": "6mGVa3SQYLkb",
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "outputId": "00471ecd-0d73-4898-e2d2-af1c8c257a86"
```

```
   },
   "execution_count": 3,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "['Hi', 'there', 'Sam!']\n"
     ]
    }
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "*`italicized text`*## 2. Use .format() to print the following string. \n",
    "\n",
    "### Output should be: The diameter of Earth is 12742 kilometers."
   ],
   "metadata": {
    "id": "GH1QBn8HP375"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "planet = \"Earth\"\n",
    "diameter = 12742"
   ],
   "metadata": {
    "id": "_ZHoml3kPqic"
   },
   "execution_count": 4,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "print(f'The diameter of {planet} is {diameter} kilometers.')"
   ],
   "metadata": {
    "id": "HyRyJv6CYPb4",
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "outputId": "4a8e1fc3-09f8-4dc0-80e9-7dea9a6fd3fe"
   },
   "execution_count": 5,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "The diameter of Earth is 12742 kilometers.\n"
     ]
    }
   ]
  },
{
 "cell_type": "markdown",
 "source": [
  "## 3. In this nest dictionary grab the word \"hello\""
 ],
 "metadata": {
  "id": "KE74ZEwkRExZ"
 }
},
{
 "cell_type": "code",
 "source": [
  "d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}"
 ],
 "metadata": {
  "id": "fcVwbCc1QrQI"
 },
 "execution_count": 6,
 "outputs": []
},
{
 "cell_type": "code",
 "source": [
  "d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}\n",
  "print(d['k1'][3]['tricky'][3]['target'][3])"
 ],
 "metadata": {
  "id": "MvbkMZpXYRaw",
  "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "outputId": "396614b9-e37f-4f10-8f17-941919b7854e"
   },
   "execution_count": 7,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "hello\n"
     ]
    }
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "# Numpy"
   ],
   "metadata": {
    "id": "bw0vVp-9ddjv"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "import numpy as np"
   ],
  },
  "metadata": {
   "id": "LLiE_TYrhA1O"
  },
  "execution_count": 8,
  "outputs": []
 },
 {
  "cell_type": "markdown",
  "source": [
   "## 4.1 Create an array of 10 zeros? \n",
   "## 4.2 Create an array of 10 fives?"
  ],
  "metadata": {
   "id": "wOg8hinbgx30"
  }
 },
 {
  "cell_type": "code",
  "source": [
   "arr=np.zeros(10)\n",
   "print(arr)"
  ],
  "metadata": {
   "id": "NHrirmgCYXvU",
   "colab": {
    "base_uri": "https://localhost:8080/"
   },
   "outputId": "904bd956-c97a-4cb3-c736-2916dc25b500"
  },
   "execution_count": 9,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]\n"
     ]
    }
   ]
  },
  {
   "cell_type": "code",
   "source": [
    "arr=np.ones(10)*5\n",
    "print(arr)"
   ],
   "metadata": {
    "id": "e4005lsTYXxx",
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "outputId": "e7d63fa1-7401-48a6-88e2-0b187a4c8a87"
   },
   "execution_count": 10,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
   "text": [
    "[5. 5. 5. 5. 5. 5. 5. 5. 5. 5.]\n"
   ]
  }
 ]
},
{
 "cell_type": "markdown",
 "source": [
  "## 5. Create an array of all the even integers from 20 to 35"
 ],
 "metadata": {
  "id": "gZHHDUBvrMX4"
 }
},
{
 "cell_type": "code",
 "source": [
  "import numpy as np\n",
  "array=np.arange(20,36,2)\n",
  "print(array)"
 ],
 "metadata": {
  "id": "oAI2tbU2Yag-",
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "outputId": "441db886-04dd-47c4-b03f-c64b03ac895a"
 },
   "execution_count": 11,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "[20 22 24 26 28 30 32 34]\n"
     ]
    }
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "## 6. Create a 3x3 matrix with values ranging from 0 to 8"
   ],
   "metadata": {
    "id": "NaOM308NsRpZ"
   }
  },
  {
   "cell_type": "code",
   "source": [
    "arr=np.arange(0,9).reshape(3,3)\n",
    "print(arr)\n"
   ],
   "metadata": {
    "id": "tOlEVH7BYceE",
    "colab": {
```

```
    "base_uri": "https://localhost:8080/"
   },
   "outputId": "47340832-e646-4736-f558-565b861a21b2"
  },
  "execution_count": 12,
  "outputs": [
   {
    "output_type": "stream",
    "name": "stdout",
    "text": [
     "[[0 1 2]\n",
     " [3 4 5]\n",
     " [6 7 8]]\n"
    ]
   }
  ]
 },
 {
  "cell_type": "markdown",
  "source": [
   "## 7. Concatinate a and b \n",
   "## a = np.array([1, 2, 3]), b = np.array([4, 5, 6])"
  ],
  "metadata": {
   "id": "hQ0dnhAQuU_p"
  }
 },
 {
  "cell_type": "code",
   "source": [
    "import numpy as np\n",
    "a=np.array([1,2,3])\n",
    "b=np.array([4,5,6])\n",
    "x=np.concatenate((a,b),axis=0)\n",
    "print(x)"
   ],
   "metadata": {
    "id": "rAPSw97aYfE0",
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "outputId": "9e69518c-b762-4d1f-85ce-5f75cc9a7729"
   },
   "execution_count": 13,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "[1 2 3 4 5 6]\n"
     ]
    }
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "# Pandas"
  ],
  "metadata": {
   "id": "dlPEY9DRwZga"
  }
 },
 {
  "cell_type": "markdown",
  "source": [
   "## 8. Create a dataframe with 3 rows and 2 columns"
  ],
  "metadata": {
   "id": "ijoYW51zwr87"
  }
 },
 {
  "cell_type": "code",
  "source": [
   "import pandas as pd\n"
  ],
  "metadata": {
   "id": "T5OxJRZ8uvR7"
  },
  "execution_count": null,
  "outputs": []
 },
 {
  "cell_type": "code",
  "source": [
   "import pandas as pd\n",
      "df = pd.DataFrame()\n",
      "print(df)"
    ],
    "metadata": {
      "id": "xNpI_XXoYhs0",
      "colab": {
        "base_uri": "https://localhost:8080/"
      },
      "outputId": "b50aa1df-8427-44e3-8335-e4310cf9b3cd"
    },
    "execution_count": 14,
    "outputs": [
      {
        "output_type": "stream",
        "name": "stdout",
        "text": [
          "Empty DataFrame\n",
          "Columns: []\n",
          "Index: []\n"
        ]
      }
    ]
  },
  {
    "cell_type": "markdown",
    "source": [
      "## 9. Generate the series of dates from 1st Jan, 2023 to 10th Feb, 2023"
    ],
    "metadata": {
```

```
   "id": "UXSmdNclyJQD"
  }
 },
 {
  "cell_type": "code",
  "source": [
   "import datetime\n",
   "import pandas as pd\n",
   "test_date = datetime.datetime.strptime(\"01-01-2023\", \"%d-%m-%Y\")\n",
   "k=41\n",
   "date_generated = pd.date_range(test_date, periods=k)\n",
   "print(date_generated.strftime(\"%d-%m-%Y\"))"
  ],
  "metadata": {
   "id": "dgyC0JhVYl4F",
   "colab": {
    "base_uri": "https://localhost:8080/"
   },
   "outputId": "012518ea-83ee-4c39-ae82-cc674b60c921"
  },
  "execution_count": 15,
  "outputs": [
   {
    "output_type": "stream",
    "name": "stdout",
    "text": [
     "Index(['01-01-2023', '02-01-2023', '03-01-2023', '04-01-2023', '05-01-2023',\n",
     "       '06-01-2023', '07-01-2023', '08-01-2023', '09-01-2023', '10-01-2023',\n",
     "       '11-01-2023', '12-01-2023', '13-01-2023', '14-01-2023', '15-01-2023',\n",
        "     '16-01-2023', '17-01-2023', '18-01-2023', '19-01-2023', '20-01-2023',\n",
        "     '21-01-2023', '22-01-2023', '23-01-2023', '24-01-2023', '25-01-2023',\n",
        "     '26-01-2023', '27-01-2023', '28-01-2023', '29-01-2023', '30-01-2023',\n",
        "     '31-01-2023', '01-02-2023', '02-02-2023', '03-02-2023', '04-02-2023',\n",
        "     '05-02-2023', '06-02-2023', '07-02-2023', '08-02-2023', '09-02-2023',\n",
        "     '10-02-2023'],\n",
        "    dtype='object')\n"
       ]
      }
     ]
    },
    {
     "cell_type": "markdown",
     "source": [
      "## 10. Create 2D list to DataFrame\n",
      "\n",
      "lists = [[1, 'aaa', 22],\n",
      "         [2, 'bbb', 25],\n",
      "         [3, 'ccc', 24]]"
     ],
     "metadata": {
      "id": "ZizSetD-y5az"
     }
    },
    {
     "cell_type": "code",
     "source": [
      "lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]"
     ],
   "metadata": {
    "id": "_XMC8aEt0llB"
   },
   "execution_count": 16,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "import pandas as pd\n",
    "lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]\n",
    "df = pd.DataFrame(lists, columns =['Sno', 'Name', 'Age'])\n",
    "print(df)"
   ],
   "metadata": {
    "id": "knH76sDKYsVX",
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "outputId": "5ac16266-e9fd-468b-aa43-6cf0030a13c8"
   },
   "execution_count": 17,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "   Sno Name  Age\n",
      "0    1  aaa   22\n",
      "1   2  bbb   25\n",
      "2   3  ccc   24\n"
     ]
    }
   ]
  }
 ]
}
```