```
{
 "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": null,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "s=\"Hi there Sam!\"\n",
    "s=s.split()\n",
    "print(s);"
  ],
  "metadata": {
    "id": "6mGVa3SQYLkb",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "f4c6110c-735d-4312-ccad-cf0138b88ea8"
  },
  "execution_count": 1,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "['Hi', 'there', 'Sam!']\n"
      ]
    }
  ]
},
{
  "cell_type": "markdown",
  "source": [
```

```
    "## 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": null,
   "outputs": []
  },
  {
   "cell_type": "code",
   "source": [
    "planet = \"Earth\"\n",
    "diameter = 12742\n",
    "planet = \"Earth\"\n",
    "diameter = 12742\n",
    "print( 'The diameter of {} is {} kilometers.' .format(planet,diameter));"
   ],
   "metadata": {
    "id": "HyRyJv6CYPb4",
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "outputId": "0aeea619-f265-4ac7-b78d-9250187c89de"
   },
   "execution_count": 2,
```

```
  "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": null,
  "outputs": []
 },
 {
  "cell_type": "code",
  "source": [
   "lst = [1,2,[3,4],[5,[100,200,['hello']],23,11],1,7]\n",
   "lst = [1,2,[3,4],[5,[100,200,['hello']],23,11],1,7]\n",
   "a=lst[3][1][2];\n",
   "print(a)"
```

```
  ],
  "metadata": {
    "id": "MvbkMZpXYRaw",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "5a119a1f-aa66-4538-c6be-6d972f008faa"
  },
  "execution_count": 3,
  "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": null,
```

```
  "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": [
    "import numpy as np\n",
    "array=np.zeros(10)\n",
    "print(\"An array of 10 zeros:\")\n",
    "print(array)\n",
    "array=np.ones(10)*5\n",
    "print(\"An array of 10 fives:\")\n",
    "print(array)"
  ],
  "metadata": {
    "id": "NHrirmgCYXvU",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "6e312b74-4962-407b-9560-776f27c85aca"
  },
  "execution_count": 4,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "An array of 10 zeros:\n",
```

```
        "[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]\n",
        "An array of 10 fives:\n",
        "[5. 5. 5. 5. 5. 5. 5. 5. 5. 5.]\n"
      ]
    }
  ]
},
{
  "cell_type": "code",
  "source": [],
  "metadata": {
    "id": "e4005lsTYXxx"
  },
  "execution_count": null,
  "outputs": []
},
{
  "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 of all the even integers from 20 to 35\")\n",
    "print(array)"
  ],
  "metadata": {
    "id": "oAI2tbU2Yag-",
    "colab": {
      "base_uri": "https://localhost:8080/"
```

```
    },
    "outputId": "b8f00c5d-6658-46aa-96e2-6fd97ed8bf7b"
   },
   "execution_count": 5,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "Array of all the even integers from 20 to 35\n",
      "[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": [
    "import numpy as np\n",
    "x =  np.arange(0, 9).reshape(3,3)\n",
    "print(x)"
   ],
   "metadata": {
    "id": "tOlEVH7BYceE",
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "outputId": "e12038d3-28e1-4376-cfaa-b4610ec5b198"
```

```
  },
  "execution_count": 6,
  "outputs": [
   {
    "output_type": "stream",
    "name": "stdout",
    "text": [
     "[[0 1 2]\n",
     " [3 4 5]\n",
     " [6 7 8]]\n"
    ]
   }
  ]
 },
 {
  "cell_type": "markdown",
  "source": [
   "## 7. Concatenate 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",
   " \n",
   "a = np.array([1, 2, 3])\n",
   "print(a)\n",
   " \n",
   "b = np.array([4, 5, 6])\n",
   "print(b)\n",
   " \n",
   "print('\\n---Result of a and b---')\n",
   "print(np.concatenate((a, b)))"
```

```
  ],
  "metadata": {
    "id": "rAPSw97aYfE0",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "38532143-9c53-4fa8-9d27-cb868d28bd26"
  },
  "execution_count": 7,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "[1 2 3]\n",
        "[4 5 6]\n",
        "\n",
        "---Result of a and b---\n",
        "[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 numpy as np\n",
    "a=np.array([1,2,3])\n",
    "b=np.array([4,5,6])\n",
    "np.concatenate((a,b),axis=0)"
  ],
  "metadata": {
    "id": "xNpI_XXoYhs0",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "9e5f2ed1-22db-415c-aee6-aa0ab602fc63"
  },
  "execution_count": 8,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "array([1, 2, 3, 4, 5, 6])"
```

```
        ]
       },
       "metadata": {},
       "execution_count": 8
     }
   ]
 },
 {
   "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 pandas as pd\n",
     "pd.date_range(start='01/01/2023',end='02/10/2023')"
   ],
   "metadata": {
     "id": "dgyC0JhVYl4F",
     "colab": {
       "base_uri": "https://localhost:8080/"
     },
     "outputId": "04366456-b497-45e1-cb04-d8af06b661d5"
   },
   "execution_count": 9,
   "outputs": [
     {
       "output_type": "execute_result",
       "data": {
         "text/plain": [
           "DatetimeIndex(['2023-01-01', '2023-01-02', '2023-01-03', '2023-01-04',\n",
           "               '2023-01-05', '2023-01-06', '2023-01-07', '2023-01-08',\n",
```

```
          "               '2023-01-09', '2023-01-10', '2023-01-11', '2023-01-12',\n",
          "               '2023-01-13', '2023-01-14', '2023-01-15', '2023-01-16',\n",
          "               '2023-01-17', '2023-01-18', '2023-01-19', '2023-01-20',\n",
          "               '2023-01-21', '2023-01-22', '2023-01-23', '2023-01-24',\n",
          "               '2023-01-25', '2023-01-26', '2023-01-27', '2023-01-28',\n",
          "               '2023-01-29', '2023-01-30', '2023-01-31', '2023-02-01',\n",
          "               '2023-02-02', '2023-02-03', '2023-02-04', '2023-02-05',\n",
          "               '2023-02-06', '2023-02-07', '2023-02-08', '2023-02-09',\n",
          "               '2023-02-10'],\n",
          "              dtype='datetime64[ns]', freq='D')"
        ]
       },
       "metadata": {},
       "execution_count": 9
     }
    ]
   },
   {
    "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": "_XMC8aEt0lIB"
```

```
    },
    "execution_count": null,
    "outputs": []
   },
   {
    "cell_type": "code",
    "source": [
     "import pandas as pd\n",
     "lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]"
    ],
    "metadata": {
     "id": "knH76sDKYsVX"
    },
    "execution_count": 10,
    "outputs": []
   }
  ]
}
```