```
{
 "cells": [
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "McSxJAwcOdZ1"
   },
   "source": [
   1. Split the string


    "s = \"Hi there Sam!\"\n",
    "print(s.split())"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {
    "id": "GH1QBn8HP375"

    "##
2. Use .format() to print the following string. \n",
    "\n",
    "### Output should be: The diameter of Earth is 12742 kilometers."
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "_ZHoml3kPqic"
   },
   "outputs": [],
   "source": [
    "planet = \"Earth\"\n",
    "diameter = 12742"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 3,
   "metadata": {
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "id": "HyRyJv6CYPb4",
    "outputId": "af4901c9-c3d4-414c-8fc0-2ba511cd50d0"
   },
   "outputs": [
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "The diameter of Earth is 12742 kilometers.\n"
     ]
```

```
 }
],
 "source": [
  "planet = \"Earth\"\n",
  "diameter = 12742\n",
  "print('The diameter of {} is {} kilometers.'.format(planet,diameter))"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "id": "KE74ZEwkRExZ"
 },
 "source": [
  "## 3. In this nest dictionary grab the word \"hello\""
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "id": "fcVwbCc1QrQI"
 },
 "outputs": [],
 "source": [
  "d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}"
 ]
},
{
 "cell_type": "code",
 "execution_count": 42,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/",
   "height": 35
  },
  "id": "MvbkMZpXYRaw",
  "outputId": "5150dd20-ac0a-4147-b33a-09aa38975bca"
 },
 "outputs": [
  {
   "data": {
    "application/vnd.google.colaboratory.intrinsic+json": {
     "type": "string"
    },
    "text/plain": [
     "'hello'"
    ]
   },
   "execution_count": 42,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
```

```
  "d = {'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}\n",
  "d['k1'][3]['tricky'][3]['target'][3]"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "id": "bw0vVp-9ddjv"
 },
 "source": [
  "# Numpy"
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "id": "LLiE_TYrhA1O"
 },
 "outputs": [],
 "source": [
  "import numpy as np"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "id": "wOg8hinbgx30"
 },
 "source": [
  "## 4.1 Create an array of 10 zeros? \n",
  "## 4.2 Create an array of 10 fives?"
 ]
},
{
 "cell_type": "code",
 "execution_count": 4,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "NHrirmgCYXvU",
  "outputId": "d168248b-7f96-4934-cabb-94ff4d6e2e6a"
 },
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]\n"
   ]
  }
 ],
 "source": [
  "import numpy as np\n",
```

```
 "array=np.zeros(10)\n",
 "print(array)"
]
},
{
"cell_type": "code",
"execution_count": 6,
"metadata": {
 "colab": {
  "base_uri": "https://localhost:8080/"
 },
 "id": "e4005lsTYXxx",
 "outputId": "75e0dafe-e1bd-47a2-eac9-2c6177a0ce5b"
},
"outputs": [
 {
  "name": "stdout",
  "output_type": "stream",
  "text": [
   "[5. 5. 5. 5. 5. 5. 5. 5. 5. 5.]\n"
  ]
 }
],
"source": [
 "import numpy as np\n",
 "array=np.ones(10)*5\n",
 "print(array)"
]
},
{
"cell_type": "markdown",
"metadata": {
 "id": "gZHHDUBvrMX4"
},
"source": [
 "## 5. Create an array of all the even integers from 20 to 35"
]
},
{
"cell_type": "code",
"execution_count": 8,
"metadata": {
 "colab": {
  "base_uri": "https://localhost:8080/"
 },
 "id": "oAI2tbU2Yag-",
 "outputId": "12668572-a93e-4a1d-86bb-4d9705667b45"
},
"outputs": [
 {
  "name": "stdout",
  "output_type": "stream",
  "text": [
   "[20 22 24 26 28 30 32 34]\n"
  ]
```

```
 }
],
 "source": [
  "import numpy as np\n",
  "array=np.arange(20,36,2)\n",
  "print(array)"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "id": "NaOM308NsRpZ"
 },
 "source": [
  "## 6. Create a 3x3 matrix with values ranging from 0 to 8"
 ]
},
{
 "cell_type": "code",
 "execution_count": 10,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "tOlEVH7BYceE",
  "outputId": "f47f8326-cef9-4021-9891-3dd14fe924c4"
 },
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "[[0 1 2]\n",
    " [3 4 5]\n",
    " [6 7 8]]\n"
   ]
  }
 ],
 "source": [
  "import numpy as np\n",
  "array=np.arange(0,9).reshape(3,3)\n",
  "print(array)"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "id": "hQ0dnhAQuU_p"
 },
 "source": [
  "## 7. Concatinate a and b \n",
  "## a = np.array([1, 2, 3]), b = np.array([4, 5, 6])"
 ]
},
{
```

```
"cell_type": "code",
"execution_count": 15,
"metadata": {
 "colab": {
  "base_uri": "https://localhost:8080/"
 },
 "id": "rAPSw97aYfE0",
 "outputId": "97d9df55-1b7f-4b91-9585-46b0ec44b4ff"
},
"outputs": [
 {
  "data": {
   "text/plain": [
    "array([1, 2, 3, 4, 5, 6])"
   ]
  },
  "execution_count": 15,
  "metadata": {},
  "output_type": "execute_result"
 }
],
"source": [
 "a = np.array([1, 2, 3])\n",
 "b = np.array([4, 5, 6])\n",
 "np.concatenate((a,b),axis=0)"
]
},
{
"cell_type": "markdown",
"metadata": {
 "id": "dlPEY9DRwZga"
},
"source": [
 "# Pandas"
]
},
{
"cell_type": "markdown",
"metadata": {
 "id": "ijoYW51zwr87"
},
"source": [
 "## 8. Create a dataframe with 3 rows and 2 columns"
]
},
{
"cell_type": "code",
"execution_count": null,
"metadata": {
 "id": "T5OxJRZ8uvR7"
},
"outputs": [],
"source": [
 "import pandas as pd\n"
]
```

```
},
{
 "cell_type": "code",
 "execution_count": 19,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "xNpI_XXoYhs0",
  "outputId": "b0595112-3027-4b6a-8fad-7bca54fba9f5"
 },
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "[[1 2]\n",
    " [3 4]\n",
    " [5 6]]\n"
   ]
  }
 ],
 "source": [
  "import pandas as pd\n",
  "col=['x','y']\n",
  "row=['a','b','c']\n",
  "array=np.arange(1,7).reshape(3,2)\n",
  "print(array)"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "id": "UXSmdNclyJQD"
 },
 "source": [
  "## 9. Generate the series of dates from 1st Jan, 2023 to 10th Feb, 2023"
 ]
},
{
 "cell_type": "code",
 "execution_count": 37,
 "metadata": {
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "id": "dgyC0JhVYl4F",
  "outputId": "5afc0b68-3e53-441f-bc8a-b21207095bd6"
 },
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "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'],\n",
   "              dtype='datetime64[ns]', freq='D') \n",
   "\n",
   " DatetimeIndex(['2023-09-02', '2023-09-03', '2023-09-04', '2023-09-05',\n",
   "               '2023-09-06', '2023-09-07', '2023-09-08', '2023-09-09',\n",
   "               '2023-09-10', '2023-09-11', '2023-09-12', '2023-09-13',\n",
   "               '2023-09-14', '2023-09-15', '2023-09-16', '2023-09-17',\n",
   "               '2023-09-18', '2023-09-19', '2023-09-20', '2023-09-21',\n",
   "               '2023-09-22', '2023-09-23', '2023-09-24', '2023-09-25',\n",
   "               '2023-09-26', '2023-09-27', '2023-09-28', '2023-09-29',\n",
   "               '2023-09-30', '2023-10-01', '2023-10-02'],\n",
   "              dtype='datetime64[ns]', freq='D') \n",
   "\n",
   "\n"
  ]
 }
],
"source": [
 "import pandas as pd\n",
 "dRan1 = pd.date_range(start ='1-1-2023', periods = 31)\n",
 "dRan2 = pd.date_range(end ='10-2-2023', periods = 31)  \n",
 "print(dRan1, \"\\n\\n\", dRan2,\"\\n\\n\")"
]
},
{
"cell_type": "markdown",
"metadata": {
 "id": "ZizSetD-y5az"
},
"source": [
 "## 10. Create 2D list to DataFrame\n",
 "\n",
 "lists = [[1, 'aaa', 22],\n",
 "         [2, 'bbb', 25],\n",
 "         [3, 'ccc', 24]]"
]
},
{
"cell_type": "code",
"execution_count": null,
"metadata": {
 "id": "_XMC8aEt0llB"
},
"outputs": [],
"source": [
 "lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]"
]
},
{
```

```
  "cell_type": "code",
  "execution_count": 41,
  "metadata": {
   "colab": {
    "base_uri": "https://localhost:8080/"
   },
   "id": "knH76sDKYsVX",
   "outputId": "235acc76-82bc-4a7e-bbc8-c4584496ec1d"
  },
  "outputs": [
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "   Tag name  age\n",
     "0    1  aaa   22\n",
     "1    2  bbb   25\n",
     "2    3  ccc   24\n"
    ]
   }
  ],
  "source": [
   "lists = [[1, 'aaa', 22], [2, 'bbb', 25], [3, 'ccc', 24]]\n",
   "df = pd.DataFrame(lists, columns =['Tag', 'name','age']) \n",
   "print(df )"
  ]
 }
],
"metadata": {
 "colab": {
  "collapsed_sections": [],
  "provenance": []
 },
 "kernelspec": {
  "display_name": "Python 3 (ipykernel)",
  "language": "python",
  "name": "python3"
 },
 "language_info": {
  "codemirror_mode": {
   "name": "ipython",
   "version": 3
  },
  "file_extension": ".py",
  "mimetype": "text/x-python",
  "name": "python",
  "nbconvert_exporter": "python",
  "pygments_lexer": "ipython3",
  "version": "3.9.12"
 }
},
"nbformat": 4,
"nbformat_minor": 1
}
```