```
{
  "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": 1,
 "outputs": []
},
{
 "cell_type": "code",
 "source": [
  "x=s.split()\n",
    "print(x)"
   ],
   "metadata": {
    "id": "6mGVa3SQYLkb",
    "outputId": "a647f9ab-ccf4-4098-93da-5da57aef359c",
    "colab": {
     "base_uri": "https://localhost:8080/"
    }
   },
   "execution_count": 2,
   "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": 3,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "txt=\"The diameter of {} is {} kilometer.\".format(planet,diameter)\n",
    "print(txt)"
  ],
  "metadata": {
    }
    "id": "HyRyJv6CYPb4",
    "outputId": "283ba490-0d6d-4188-e441-587889a9f2ff",
    "colab": {
     "base_uri": "https://localhost:8080/"
    }
   },
   "execution_count": 4,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "The diameter of Earth is 12742 kilometer.\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": 5,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "d['k1'][3]['tricky'][3]['target'][3]"
  ],
  "metadata": {
    "id": "MvbkMZpXYRaw",
    "outputId": "edb50186-16ab-4f57-910e-105681f49efa",
    "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 36
    }
  },
   }
   "execution_count": 6,
   "outputs": [
    {
     "output_type": "execute_result",
     "data": {
      "text/plain": [
       "'hello'"
      ],
      "application/vnd.google.colaboratory.intrinsic+json": {
       "type": "string"
      }
     },
     "metadata": {},
     "execution_count": 6
    }
   ]
  },
  {
   "cell_type": "markdown",
   "source": [
    "# Numpy"
   ],
   "metadata": {
    "id": "bw0vVp-9ddjv"
   }
```

```
},
{
 "cell_type": "code",
 "source": [
  "import numpy as np"
 ],
 "metadata": {
  "id": "LLiE_TYrhA1O"
 },
 "execution_count": 7,
 "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": [
    "array = np.zeros(10)\n",
    "print(\"An array of 10 zeros:\")\n",
    "print(array)"
   ],
   "metadata": {
    "id": "NHrirmgCYXvU",
    "outputId": "ab8fedfd-9a37-4e2a-e985-f55a4f4b86a4",
    "colab": {
     "base_uri": "https://localhost:8080/"
    }
   },
   "execution_count": 8,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "An array of 10 zeros:\n",
      "[0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]\n"
     ]
    }
   ]
  },
  {
   "cell_type": "code",
   "source": [
    "array=np.ones(10)*5\n",
    "print(\"An array of 10 fives:\")\n",
    "print(array)"
   ],
   "metadata": {
    "id": "e4005lsTYXxx",
    "outputId": "bb0cc07d-f1b7-4682-c4f7-c067b8fcc2e0",
    "colab": {
     "base_uri": "https://localhost:8080/"
    }
   },
   "execution_count": 9,
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "An array of 10 fives:\n",
      "[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": [
    "array =np.arange(20,35,2)\n",
    "print(\"Array of all the even integers from 20 to 35\")\n",
    "print(array)"
   ],
   "metadata": {
    "id": "oAI2tbU2Yag-",
    "outputId": "74d46e4b-60fb-42ab-cc98-dd9e8ae7e8df",
    "colab": {
     "base_uri": "https://localhost:8080/"
    }
   },
   "execution_count": 10,
   "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": [
    "x=np.arange(0,9).reshape(3,3)\n",
    "print(x)"
   ],
   "metadata": {
    "id": "tOlEVH7BYceE",
    "outputId": "718dd269-3089-462f-9b33-259e2378f77b",
    "colab": {
     "base_uri": "https://localhost:8080/"
    }
   },
   "execution_count": 11,
   "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": [
    "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",
    "outputId": "0ba6fcb7-bfc2-4b17-f4bf-655a2a693ffa",
    "colab": {
     "base_uri": "https://localhost:8080/"
    }
   },
   "execution_count": 12,
   "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": 13,
  "outputs": []
 },
 {
  "cell_type": "code",
  "source": [
   "data=[{'a':13,'b':22},{'a':64,'b':15},{'a':44,'b':47}]\n",
   "df=pd.DataFrame(data)\n",
   "print (df)"
  ],
  "metadata": {
   "id": "xNpI_XXoYhs0",
   "outputId": "707f5961-986c-4832-9c38-60a27ec21da7",
   "colab": {
    "base_uri": "https://localhost:8080/"
   }
  },
  "execution_count": 22,
  "outputs": [
     {
      "output_type": "stream",
      "name": "stdout",
      "text": [
       "    a   b\n",
       "0  13  22\n",
       "1  64  15\n",
       "2  44  47\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": [
    "a=pd.date_range(start='1/1/2023',end='10/2/2023')\n",
  "print(a)"
 ],
 "metadata": {
  "id": "dgyC0JhVYl4F",
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "outputId": "db806eb0-e26a-48a1-93d5-1c00c31985e2"
 },
 "execution_count": 21,
 "outputs": [
  {
   "output_type": "stream",
   "name": "stdout",
   "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',\n",
    "               ...\n",
    "               '2023-09-23', '2023-09-24', '2023-09-25', '2023-09-26',\n",
    "               '2023-09-27', '2023-09-28', '2023-09-29', '2023-09-30',\n",
    "               '2023-10-01', '2023-10-02'],\n",
    "              dtype='datetime64[ns]', length=275, freq='D')\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": 17,
  "outputs": []
```

```
},
{
 "cell_type": "code",
 "source": [
  "df=pd.DataFrame(lists)\n",
  "print(df)"
 ],
 "metadata": {
  "id": "knH76sDKYsVX",
  "outputId": "0e097db5-1366-48ab-cb5c-09c3fcbc24ec",
  "colab": {
   "base_uri": "https://localhost:8080/"
  }
 },
 "execution_count": 18,
 "outputs": [
  {
   "output_type": "stream",
   "name": "stdout",
   "text": [
    "   0    1   2\n",
    "0  1  aaa  22\n",
    "1  2  bbb  25\n",
    "2  3  ccc  24\n"
   ]
```

```
        }
      ]
    }
  ]
}
```