```
{
  "cells": [
    {
      "cell_type": "markdown",
      "metadata": {
        "id": "McSxJAwcOdZ1"
      },
      "source": [
        "# Basic Python"
      ]
    },
    {
      "cell_type": "markdown",
      "metadata": {
        "id": "CU48hgo4Owz5"
      },
      "source": [
        "## 1. Split this string"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {
        "id": "s07c7JK7Oqt-"
      },
      "outputs": [],
      "source": [
        "s = \"Hi there Sam!\""
      ]
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {
        "id": "6mGVa3SQYLkb",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "748a3e02-71e4-433e-dedd-824abcc31980"
      },
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
        "['Hi', 'there', 'Sam!']\n"
      ]
    }
  ],
  "source": [
    "s = \"Hi there Sam!\"\n",
    "n=s.split()\n",
    "print(n)"
  ]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "GH1QBn8HP375"
  },
  "source": [
    "## 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": null,
  "metadata": {
    "id": "HyRyJv6CYPb4",
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "outputId": "28d17ce1-3b83-4e22-edda-5ff64a976ed0"
  },
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "The diameter of Earth is 12742 kilometers\n"
      ]
    }
  ],
  "source": [
    "planet = \"Earth\"\n",
    "diameter = 12742\n",
    "star=\"The diameter of {p} is {k} kilometers\"\n",
    "print(star.format(p=planet,k=diameter))"
  ]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "KE74ZEwkRExZ"
  },
  "source": [
    "## 3. In this nest dictionary grab the word \"hello\""
  ]
},
{
  "cell_type": "markdown",
  "source": [],
  "metadata": {
    "id": "69Vm0vJ-SwM7"
  }
},
{
  "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": null,
  "metadata": {
    "id": "MvbkMZpXYRaw",
    "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 36
    },
    "outputId": "7dd4c7a3-3035-48be-bef3-f7ca3dd8f978"
  },
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "'hello'"
        ],
        "application/vnd.google.colaboratory.intrinsic+json": {
          "type": "string"
        }
      },
      "metadata": {},
      "execution_count": 10
    }
  ],
  "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",
      "source": [
        "array=np.zeros(10)\n",
        "print(\"An array of 10 zeros\")\n",
        "print(array)"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "ycMFYZU_Tykc",
        "outputId": "5de2efea-d038-4119-a58d-b2e319846e8d"
      },
      "execution_count": null,
      "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",
      "execution_count": null,
      "metadata": {
        "id": "NHrirmgCYXvU",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "5f993b37-a5bd-4dc9-d36f-fcb026ca517c"
      },
      "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "An array of 10 fives\n",
      "[5. 5. 5. 5. 5. 5. 5. 5. 5. 5.]\n"
     ]
    }
   ],
   "source": [
    "array=np.ones(10)*5\n",
    "print(\"An array of 10 fives\")\n",
    "print(array)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "e4005lsTYXxx"
   },
   "outputs": [],
   "source": []
  },
  {
   "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": null,
   "metadata": {
    "id": "oAI2tbU2Yag-",
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "outputId": "0d3ea60b-52a7-4095-daf0-34f060df803d"
   },
   "outputs": [
    {
     "output_type": "stream",
   "name": "stdout",
   "text": [
    "[20 22 24 26 28 30 32 34]\n"
   ]
  }
 ],
 "source": [
  "a=np.arange(20,35,2)\n",
  "print(a)"
 ]
},
{
 "cell_type": "markdown",
 "metadata": {
  "id": "NaOM308NsRpZ"
 },
 "source": [
  "## 6. Create a 3x3 matrix with values ranging from 0 to 8"
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "id": "tOlEVH7BYceE",
  "colab": {
   "base_uri": "https://localhost:8080/"
  },
  "outputId": "cb1705e5-95c7-40cf-a9b7-0c4eb134bd13"
 },
 "outputs": [
  {
   "output_type": "stream",
   "name": "stdout",
   "text": [
    "[[0 1 2]\n",
    " [3 4 5]\n",
    " [6 7 8]]\n"
   ]
  }
 ],
 "source": [
  "x=np.arange(0,9).reshape(3,3)\n",
  "print(x)"
 ]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "hQ0dnhAQuU_p"
  },
  "source": [
    "## 7. Concatenate a and b \n",
    "## a = np.array([1, 2, 3]), b = np.array([4, 5, 6])"
  ]
},
{
  "cell_type": "code",
  "source": [
    "import numpy as np\n"
  ],
  "metadata": {
    "id": "h-VgrLqRho0B"
  },
  "execution_count": 9,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "a=np.array([1,2,3])\n",
    "b=np.array([4,5,6])"
  ],
  "metadata": {
    "id": "cn3WDnhchxP2"
  },
  "execution_count": 10,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "np.concatenate((a, b))"
  ],
  "metadata": {
    "id": "Rh2bSqxUhy2V",
    "outputId": "62c8ba45-6072-45a5-aef8-7bf44e41debc",
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },
  "execution_count": 11,
      "outputs": [
        {
          "output_type": "execute_result",
          "data": {
            "text/plain": [
              "array([1, 2, 3, 4, 5, 6])"
            ]
          },
          "metadata": {},
          "execution_count": 11
        }
      ]
    },
    {
      "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",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "44dd7db6-8aff-4fe1-a49f-f332bafd1b1e"
      },
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "     Name  Age\n",
            "0   vamsi   10\n",
            "1  mahesh   20\n",
            "2     sai   30\n"
          ]
        }
      ],
      "source": [
        "import pandas as pd\n",
        "data=[['vamsi',10],['mahesh',20],['sai',30]]\n",
        "a=pd.DataFrame(data,columns=['Name','Age',])\n",
        "print(a)\n"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {
        "id": "xNpI_XXoYhs0"
      },
      "outputs": [],
      "source": []
    },
    {
      "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": null,
      "metadata": {
        "id": "dgyC0JhVYl4F",
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "outputId": "85f2c3c0-8f54-4726-8ac1-928665e36457"
      },
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "[datetime.datetime(2023, 1, 1, 0, 0), datetime.datetime(2023, 1, 2, 0, 0),
datetime.datetime(2023, 1, 3, 0, 0), datetime.datetime(2023, 1, 4, 0, 0),
datetime.datetime(2023, 1, 5, 0, 0), datetime.datetime(2023, 1, 6, 0, 0),
datetime.datetime(2023, 1, 7, 0, 0), datetime.datetime(2023, 1, 8, 0, 0),
datetime.datetime(2023, 1, 9, 0, 0), datetime.datetime(2023, 1, 10, 0, 0),
datetime.datetime(2023, 1, 11, 0, 0), datetime.datetime(2023, 1, 12, 0, 0),
datetime.datetime(2023, 1, 13, 0, 0), datetime.datetime(2023, 1, 14, 0, 0),
datetime.datetime(2023, 1, 15, 0, 0), datetime.datetime(2023, 1, 16, 0, 0),
datetime.datetime(2023, 1, 17, 0, 0), datetime.datetime(2023, 1, 18, 0, 0),
datetime.datetime(2023, 1, 19, 0, 0), datetime.datetime(2023, 1, 20, 0, 0),
datetime.datetime(2023, 1, 21, 0, 0), datetime.datetime(2023, 1, 22, 0, 0),
datetime.datetime(2023, 1, 23, 0, 0), datetime.datetime(2023, 1, 24, 0, 0),
datetime.datetime(2023, 1, 25, 0, 0), datetime.datetime(2023, 1, 26, 0, 0),
datetime.datetime(2023, 1, 27, 0, 0), datetime.datetime(2023, 1, 28, 0, 0),
datetime.datetime(2023, 1, 29, 0, 0), datetime.datetime(2023, 1, 30, 0, 0),
datetime.datetime(2023, 1, 31, 0, 0), datetime.datetime(2023, 2, 1, 0, 0),
datetime.datetime(2023, 2, 2, 0, 0), datetime.datetime(2023, 2, 3, 0, 0),
datetime.datetime(2023, 2, 4, 0, 0), datetime.datetime(2023, 2, 5, 0, 0),
datetime.datetime(2023, 2, 6, 0, 0), datetime.datetime(2023, 2, 7, 0, 0),
datetime.datetime(2023, 2, 8, 0, 0), datetime.datetime(2023, 2, 9, 0, 0),
datetime.datetime(2023, 2, 10, 0, 0)]\n"
          ]
        }
      ],
      "source": [
        "from datetime import datetime,timedelta\n",
        "\n",
        "def date_range(start,end):\n",
        "    delta=end - start\n",
        "    days= [start + timedelta(days=i) for i in range(delta.days + 1)]\n",
        "    return days\n",
        "start_date=datetime(2023,1,1)\n",
        "end_date=datetime(2023,2,10)\n",
        "print(date_range(start_date, end_date))"
      ]
    },
    {
      "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": null,
   "metadata": {
    "id": "knH76sDKYsVX",
    "colab": {
     "base_uri": "https://localhost:8080/"
    },
    "outputId": "c7a77eca-6958-4c5f-ee2f-018cfd1f94a6"
   },
   "outputs": [
    {
     "output_type": "stream",
     "name": "stdout",
     "text": [
      "   Number FName  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=['Number','FName','Age'])\n",
    "print(df)\n"
   ]
  }
 ],
 "metadata": {
  "colab": {
   "collapsed_sections": [],
   "provenance": []
  },
  "kernelspec": {
   "display_name": "Python 3",
   "name": "python3"
  },
  "language_info": {
   "name": "python"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 0
```