



PYQGIS

PYTHON PARA QGIS

55 HORAS





FORMACIÓN ESPECÍFICA ADAPTADA

Imasgal desarrolla formación específica para un alumnado cuyo perfil está relacionado con el sector de la ingeniería, la arquitectura o el diseño. Contamos con un equipo de docentes expertos en cada área de formación. Los cursos se gestionan mediante una plataforma de formación donde los alumnos encuentran foros, acceso a las webinars, materiales, videotutoriales etc.

Imasgal realiza cursos centrando sus esfuerzos en tres características clave:

WEBINARS + E-LEARNING

Las clases en tiempo real a través de internet mejoran la interacción docente alumno. Si el alumno no puede asistir, puede visualizarlas en diferido. La parte e-learning permite al alumno desarrollar prácticas y proyectos estando tutorizado en todo momento. ■

MATERIALES ÚTILES

Trabajamos para desarrollar manuales de alta calidad que sirvan para el seguimiento del curso y posterior guía de consulta. Como complemento realizamos videotutoriales y recursos complementarios, todo orientado a maximizar el rendimiento del alumno. ■

POST-FORMACIÓN

Una vez finalizado el curso, los alumnos tienen acceso a un aula virtual de post-formación durante 2 meses, que contiene todos los contenidos del curso (vídeos grabados de webinars, videotutoriales y recursos). ■

PHYTON

EN QGIS



1

PRESENTACIÓN

El curso **Python en QGIS (PyQGIS)** capacita al alumno para crear aplicaciones de desarrollo GIS utilizando el lenguaje de programación Python, y visualizar los resultados obtenidos por consola en el entorno de QGIS.

Una vez finalizado el curso el alumno es capaz de desarrollar a través de lenguaje Python herramientas y análisis aplicados a su propio GIS, integrar, procesar, analizar y editar la información GIS, visualizando todo ello en el entorno QGIS.



TITULACIÓN:

Certificado acreditativo de superación del curso.



DURACIÓN:

55h



MODALIDADES DISPONIBLES (*):

Online (30 h webinar + 25 h e-learning)



PRECIO:

Consultar precios en web.



MATERIAL:

Manuales, recursos, videotutoriales, grabaciones webinars, ejercicios complementarios.



FORMACIÓN BONIFICADA:

Curso bonificable en las cotizaciones de la Seguridad Social.

(* Modalidad presencial disponible para grupos o empresas. Solicitar información en formación@imasgal.com

2 OBJETIVOS: PHYTON EN QGIS

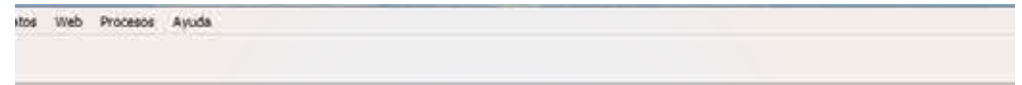
El curso Python en QGIS (PyQGIS) capacita al alumno para manejar de forma eficiente y fluida QGIS a través de su consola Python y el desarrollo de aplicaciones.

Objetivos específicos:

- Interactuar con entorno python desde QGIS.
- Crear tus propias bibliotecas de información geográfica.
- Agregar nuevos complementos.
- Importar y exportar información vectorial y raster.
- Acceder, analizar, procesar, integrar, crear y editar capas.
- Asignar CRS y metadatos a las capas.
- Diseñar y exportar mapas más personalizados.
- Realizar operaciones y geoprosesos.
- Generar tablas e informes.
- Desarrollar tus propias aplicaciones GIS.
- Publicación en QGIS Cloud.

3 DESTINATARIOS

Debido a la amplitud de uso de los GIS, este curso está dirigido a estudiantes y/o profesionales del ámbito de las Geo-ciencias, tanto desde el punto de vista de una licenciatura (Geografía, Geología, Biología, Ciencias Ambientales, Gestión Ambiental, etc.) como de una ingeniería (Arquitectura, Topografía, Civil, Forestal, Ambiental, Agrícola, Agronómica, etc.), es decir, cualquier persona que desee ir más allá en los entornos GIS y poder automatizar o diseñar sus propias aplicaciones.



```

1 # Personaliza la Aplicacion
2 from PyQt4.QtGui import *
3 from qgis.core import *
4 from qgis.utils import iface
5 from PyQt4.QtCore import *
6 from PyQt4.QtGui import *
7 app = QDialog()
8 titulo = "Introduce una capa"
9 label = "Nombre Capa: carretera o CCAA"
10 mode = QLineEdit.Normal
11 default = "<Nombre de la capa>"
12 text, True = QDialog.getText(app, titulo, label, mode, default)
13 print text
14 -if text == 'carretera':
15     carretera = QgsVectorLayer(r"C:\Data\Python_QGIS_Scripts\Recursos
16     print u"Podrás ver la capa cargada en la ventana de mapa."
17 - if not carretera.isValid():
18     | print "La capa no se ha podido cargar", carretera.name()
19     QgsMapLayerRegistry.instance().addMapLayers([carretera])
20 -elif text == 'CCAA':
21     CCAA = QgsVectorLayer(r"C:\Data\Python_QGIS_Scripts\Recursos\Vect
22     print u"Podrás ver la capa cargada en la ventana de mapa."
23 - if not CCAA.isValid():
24     | print "La capa no se ha podido cargar", CCAA.name()
25     QgsMapLayerRegistry
  
```

**PERSONALIZAR APLICACIONES
MEDIANTE CÓDIGO PYTHON-QGIS**

El curso **Python en QGIS** se imparte en modalidad online con 30 h en clases webinar + 25 h e-learning.

/ CLASES WEBINAR

Asistir a clases webinars significa que las clases se desarrollan en tiempo real a través de internet con una interacción total docente - alumno.

Durante las sesiones webinar el alumno visualiza el ordenador del docente mientras se realizan las explicaciones. Por su parte, el docente visualiza el ordenador del alumno para su seguimiento, mientras se realizan los ejercicios prácticos propuestos.

Las clases son grabadas y el alumno puede verlas en diferido.

/ PARTE E-LEARNING

En esta parte el alumno realiza prácticas relacionadas con la materia, estando tutorizado por los docentes mediante foros y webinars individuales.

/ POST-FORMACIÓN

Una vez finalizado el curso, los alumnos tienen acceso a un aula virtual de post-formación, con todos los contenidos del curso (vídeos grabados de webinars, videotutoriales y recursos).

PLATAFORMA DE FORMACIÓN

Todo el curso está gestionado mediante una plataforma de formación online. En la misma se gestionan los contenidos del curso así como se da soporte a las dudas que surgen durante el curso fuera de las clases webinars.

Foro

El foro es la herramienta de comunicación principal durante la parte e-learning y es utilizado por docentes y alumnos para solventar dudas y compartir experiencias e ideas.

Materiales

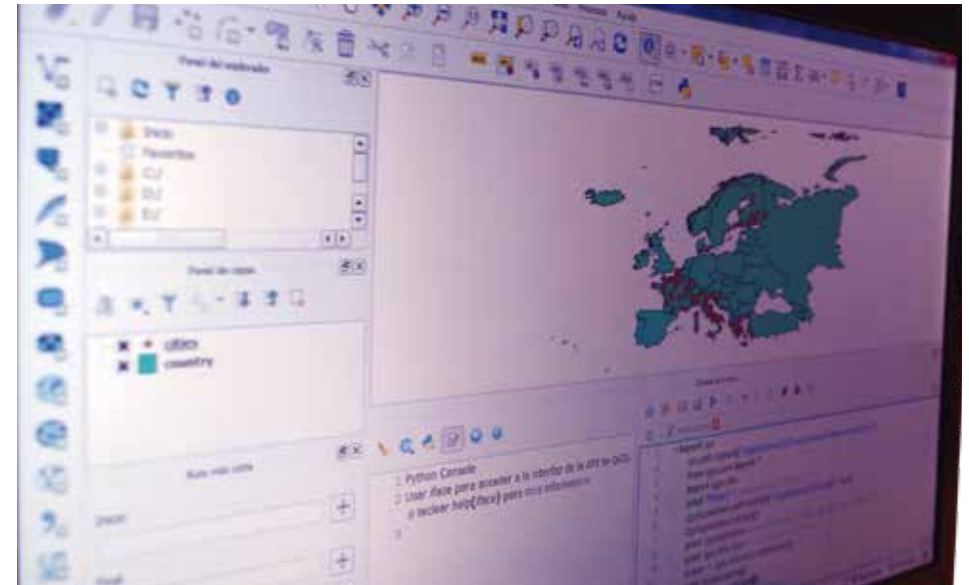
El alumno dispondrá de manuales de contenidos completos, videotutoriales, recursos complementarios y grabaciones de webinars.

Prácticas y proyectos

La prácticas y proyectos que el alumno realiza en la parte e-learning son acompañados de documentos de apoyo y recursos que facilitan su desarrollo.

Seguimos aquí

En la parte e-learning seguimos estando presentes de forma directa mediante tutorías con webinars. Queremos que la interacción docente alumno sea directa durante todo el curso.



```

$ python
Python 2.6.5 (r265:79063, Apr 16 2010, 13:09:56)
[GCC 4.4.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> print "Python is " + "a programming language"
Python is a programming language
>>> s="Python is "
>>> p="a programming language"
>>> print s+p
Python is a programming language
>>> 

```

PROGRAMA FORMATIVO: PYGIS : PYTHON PARA QGIS

1. INTRODUCCIÓN Y PRIMER CONTACTO

PYTHON – QGIS.

- 1.1 Introducción e instalación QGIS y complementos.
- 1.2 Entorno Python – QGIS.
- 1.3 Configuración de entorno.
- 1.4 Extensiones comprensibles por Python – QGIS.
- 1.5 Plugin, API y Binding.
- 1.6 Bibliografía importante y dónde encontrarla.
- 1.7 Python – QGIS: Ajustes de la consola.
- 1.8 Recomendaciones.
- 1.9 Scripts Python: variables, palabras clave, operaciones, listas y tuplas.
- 1.10 Clases y objetos.
- 1.11 Elementos más importantes de un código y su estructura.

2. CARGAR UN PROYECTO Y ACCESO A LA INFORMACIÓN.

- 2.1 Conceptos teóricos: Proyecto y sus parámetros.
- 2.2 Cargar un proyecto y asignar parámetros.
- 2.3 Conceptos teóricos: Capas y características.
- 2.4 Cargar capa vector. Información y características.
- 2.5 Cargar capa raster. Información y características.
- 2.6 Utilizar biblioteca Open Source GIS.
- 2.7 Sistemas de Coordenadas: CRS y Proyecciones.

3. TRABAJAR CON CAPAS VECTOR.

- 3.1 Acceder a los atributos.
- 3.2 Seleccionar atributos.
- 3.3 Editar, añadir, modificar, eliminar y salvar atributos.
- 3.4 Simbología.
- 3.5 Cuestionario teórico Tema 3.

4. TRABAJAR CON CAPAS RÁSTER.

- 4.1 Información de la capa.
- 4.2 Referencia a objetos (íface).
- 4.3 Convertir raster.

- 4.4 Actualizar eliminar y salvar una capa.
- 4.5 Estilos y simbología.

5. GEOMETRÍA.

- 5.1 Acceder a la geometría.
- 5.2 Construir geometría.
- 5.3 Biblioteca GEOS.

6. MAQUETACIÓN E IMPRESIÓN DE MAPA.

- 6.1 Añadir capas al mapa.
Agregar mapa base.
Herramientas de mapa.
Añadir elementos del mapa (Escala, flecha norte, cuadrícula...).
- Representar, maquetar y exportar.

7. ANÁLISIS, ALMACENAMIENTO, PLUGINS Y HERRAMIENTAS.

- 7.1 Almacenamiento de variables
- 7.2 Expresiones: Analizar y evaluar.
- 7.3 Plugins, complementos y repositorio.
- 7.4 Crear herramientas interactivas.
- 7.5 Errores más comunes y posibles soluciones.

8. PUBLICACIÓN DE UN MAPA WEB.

- 8.1 Plugins para publicación web.
- 8.2 Registro y carga de información.
- 8.3 Visualización web.

```

def are_connected(self, user1, user2):
    if self.filter(from_user=user1, to_user=user2).count() > 0:
        return True
    if self.filter(from_user=user2, to_user=user1).count() > 0:
        return True
    return False

def remove(self, user1, user2):
    """
    Deletes proper object regardless of the order of users in graph
    """
    connection = self.filter(from_user=
    if not connection:

```



```

#!/usr/bin/env python
# -*- coding: utf-8 -*-
# Script cargar capa dxf a proyecto
uri="C:\Data\Python_QGIS_Scripts\Tema_2\datos\carreteras_dxf\carreteras_dxf.dxf"
# URI Uniform Resource Identifier
capaVec = QgsVectorLayer(uri, u"Carreteras", "Carreteras")
if capaVec.isValid():
    print u"Podrás visualizar la capa en la ventana de mapa"
if not capaVec.isValid():
    print "Ups... La capa " + capaVec.name() + " no se pudo cargar. Revisa la ruta"
QgsMapLayerRegistry.instance().addMapLayer(capaVec)

```

Durante el curso se realizan proyectos reales con los que se pone en práctica los conocimientos adquiridos.

WWW.IMASGAL.COM

```
"""
Set of connections for user.
"""
def are_connected(self, user1, user2):
    if self.filter(from_user=user1, to_user=user2).count() > 0:
        return True
    if self.filter(from_user=user2, to_user=user1).count() > 0:
        return True
    return False

def remove(self, user1, user2):
    """
    Deletes proper object regardless of the order of
    connection = self.filter
    """
```