

# APOSTILA DE INTRODUÇÃO À PROGRAMAÇÃO NO MOODLE

## SUMÁRIO

Descrição do Curso.	2
Metas e visão geral do curso.	3
Principais Objetivos do Curso.	3
Trabalho final	4
Participação classe.	4
Queda de Curso.	5
Alojamento Deficiência estudante.	5
Questionários.	5
Atribuições.	5
Participação.	5
Projeto Final	5
Unidade 1: Getting Started e Conceitos Gerais.	6
Unidade 2: Criando e trabalhando no ambiente de desenvolvimento.	9
Acessibilidade Web.	12
Unidade 3: gerenciamento de código fonte.	13
Unidade 5 Parte A: Estrutura de Diretórios do Moodle.	19
O / lang.	21
O diretório / mod.	22
Unidade 5 Parte B: Variáveis Globais.	23
Globals Moodle.	23
\$ CFG..	23
\$ SITE.	27
Unidade C Parte 5: Bibliotecas Moodle.	30
Unidade 6: Visão geral de práticas de programação.	38
Erros na programação.	38
Cross-Site Scripting (XSS)	40
O que é XSS?.	40
Como podemos proteger contra XSS no Moodle?.	42
Quais são as injeções de SQL?.	43
Unidade 7 – Parte A – Bloco Básico.	50

Unidade 7 – Parte B – Formulários.	58
Unidade 7 – Parte C – Refinar a Interface de Usuário.	71
Unidade 7 – Parte D – Adicionar recursos de edição.	78
Unidade 7 – Parte E – definir um papel para o Bloco.	85
Apêndice A: Convenções de nomenclatura Moodle.	92
Unidade 7 – Apêndice B – O Editor XMLDb.	92
Apêndice D: A atualização de um Bloco.	94
Unidade 8: Teste de Métodos.	99
Unidade 10 – Programação de um Bloco Moodle.	115

## Descrição do Curso

Este curso irá fornecer uma introdução aos conhecimentos necessários para criar um bloco de Moodle. Os participantes vão examinar as diretrizes Moodle aceita codificação, incluindo questões de segurança e documentação. Os materiais do curso são projetados para Moodle versão 1.8.

## Metas e visão geral do curso

Este curso totalmente on-line começa com uma visão geral de alguns conceitos gerais e culmina com um projeto final de um bloco de programação Moodle. Os participantes devem esperar para colocar uma média de 10 horas para cada unidade do curso para completar leituras e atividades atribuídas. Além das tarefas específicas de programação e testes, outras atividades incluem a participação em discussões e oferecer feedback aos colegas de classe e facilitador.

## Principais Objetivos do Curso

- \* Junte-se e explore a comunidade on-line Moodle
- \* Reconhecer as questões de acessibilidade web para os indivíduos com diferentes capacidades físicas e limitações de hardware
- \* Comentário de uma variedade de ferramentas de edição para testes, depuração e desenvolvimento PHP
- \* Use o Eclipse para desenvolvimento PHP
- \* Iniciar e parar o Apache e serviços MySQL usando XAMPP
- \* Saída de código fonte Moodle apropriadas para o ambiente de desenvolvimento
- \* Moodle Instale no ambiente de desenvolvimento local
- \* Configure Moodle com um bloco de terceiros já existentes e módulo
- \* Rever a estrutura do diretório Moodle, reconhecendo os arquivos de biblioteca mais comum acessados por desenvolvedores
- \* Identificar as funções mais usadas no API Moodle
- \* Aplicar estilo de codificação Moodle e diretrizes
- \* Aplicar pré-requisito PHP e habilidades de programação SQL para recriar um bloco Moodle pré-definidos
- \* Compor a documentação pertinente em todas as fases do processo de programação
- \* Identificar uma função para testar em um bloco simples e desenvolver um teste para ele
- \* Detectar segurança, acessibilidade, documentação, orientação e erros de codificação em uma amostra de código
- \* Design e produzir um documento de requisitos completo para um projeto de programação com os processos de negócio identificados e os requisitos funcionais
- \* Criar um programa com uma sequência lógica de comandos executáveis, resultando em um bloco Moodle funcionamento

\* Reveja as etapas para liberação do código para a comunidade Moodle Moodle.

## Pré-requisitos

Participantes do curso deverão ter:

- \* Nível básico experiência nível de programação, incluindo PHP e SQL
- \* Conhecimento básico de SQL geral – onde as cláusulas, por fim, o grupo por, diferentes tipos de junção, etc
- \* A familiaridade com a interface de usuário do Moodle com privilégios de edição
- \* A experiência passada de instalar software
- \* Experiência com programação orientada a objeto
- \* Compreensão fundamental de XHTML e HTML tags e atributos
- \* A consciência do Subversion (SVN) ou Concurrent Versions System (CVS) em um ambiente local
- \* WinZip ou outro programa para descompactar arquivos e gerenciar tarballs
- \* Privilégios de administrador (pode instalar o software em) um computador moderno (não mais de 6 anos de idade), com uma tela de resolução 1024×768, ligação à Internet de alta velocidade, e um mínimo de 2Gb de RAM.
- \* Windows XP ou Vista sistema operacional (Linux ou Mac OS pode ser usado, mas as instruções e ilustrações são para o sistema operacional Windows)

## Honestidade Acadêmica

A menos que especificamente mencionado em direções para uma atividade particular, todas as atribuições apresentadas neste curso são para ser concluída de forma independente. Plágio batota, e outras formas de desonestidade acadêmica não serão aceitos nesta classe. Referem-se à Universidade Estadual Humboldt Política de Honestidade Acadêmica para mais detalhes.

## Trabalho final

Um ambiente de programação profissional depende de prazos do projeto. Muitas vezes trabalham outros membros da equipe é afetado quando um companheiro de equipe não cumprir um prazo. Participantes de classe são uma equipe de aprendizagem. Como resultado, a apresentação de trabalho atribuídos até à data limite é aspecto importante deste curso. Não justificadas trabalhar até tarde será penalizada de 50%. Atribuições atrasados não serão aceitos sem penalidade, quando dispensado pelo facilitador de antecedência.

## Comentários sobre Atribuições.

Os participantes podem esperar feedback sobre atribuições dentro de 3 dias da data de vencimento.

## Participação classe

Interação com os participantes outra classe é um aspecto importante da aprendizagem neste curso.

Fóruns de discussão on-line oferecem a oportunidade de compartilhar idéias e comentários.

Linguagem respeitosa é livre de palavrões e comentários inapropriados ou insulto. Comunicação desrespeitosos não serão tolerados nesta classe.

## Queda de Curso

Se você não entrar e participar activamente neste curso nas primeiras duas semanas, você será deixado com este curso.

## Alojamento Deficiência estudante

Se você é elegível para acomodações relacionadas com a deficiência como a determinação por pelo nosso Centro de Recursos campus Deficiência, entre em contato com o facilitador o mais rapidamente possível para discutir estas. Nosso campus Disability Resource Center (DRC) pode ajudá-lo a determinar a elegibilidade para acomodações e pode ser alcançado em: (707) 826-4678, (707) 826-5392 TDD. República Democrática do Congo está localizada na Casa 71 (Apartments Little) off Círculo Library.

## Avaliações e classificação

Os participantes que tomaram esta classe para uma classe serão matriculados para o crédito obrigatório / não classificação de crédito. Veja a página de Atribuição Detalhes de uma compilação de todas as avaliações para o curso.

## Questionários

Algumas Unidades em curso incluem testes para verificar a compreensão básica. Duas tentativas são permitidas.

## Atribuições

Em cada uma das unidades do curso, você vai proporcionar atividades autênticas “hands-on” experiências com os conceitos-chave da unidade. Atribuições e as discussões vão ajudar a prepará-lo para concluir com sucesso o projeto final.

## Participação

Além da participação nas discussões em aula, os participantes terão várias oportunidades para fornecer feedback aos colegas de classe e / ou o facilitador. Essas interações são cruciais não só para seu próprio aprendizado, mas o aprendizado dos outros no curso.

## Projeto Final

O projeto final é uma atividade culminante no qual cada participante completa a programação de um novo bloco. O projeto final pode ser completado individualmente ou como parte de um grupo.

## Unidade 1: Introdução e Conceitos Gerais

Bem-vindo a este curso de Introdução ao Moodle Programação desenvolvidos e oferecidos na Humboldt State University!

Desde 2004, a equipe da Humboldt State University, assumiu um papel de liderança na comunidade internacional no apoio de Moodle. Essa liderança foi reconhecida pela Andrew W. Mellon Foundation com uma doação para apoiar o desenvolvimento deste curso para expandir ainda mais nossas contribuições.

O que é o Moodle? Moodle é um software que pode ser usado para produzir cursos baseados na web e web sites para apoiar projetos colaborativos. Ele é projetado para suportar um modelo de construtivismo social da educação.

Há muitas maneiras de personalizar Moodle – tanto a nível de sistema, tanto ao nível do curso. A interface do Moodle consiste em “seções” e “blocos”. O conteúdo de “seções” é gerido pelo Professor ou Design Instrucional do curso, adicionando textos e/ou atividades. “Blocos” são criados por programadores e instalado no nível do sistema como opções o professor pode selecionar a partir de personalizar a sala de aula online. Blocos tipicamente aparecem nas colunas direita e esquerda da página do curso principal, enquanto seções aparecem abaixo a coluna central. Alguns blocos são uma parte “padrão” do núcleo do Moodle e outros são as opções disponíveis de terceiros na comunidade Moodle. Exemplos de padrão “blocos” selecionado para ser utilizado neste curso incluem o “Calendário” e “Pessoas”.

Os “módulos” do Moodle geralmente contêm o código de controle das atividades de ensino, como um quiz ou um fórum de discussão.

Moodle é fornecido livremente como software “open source”. Este modelo de distribuição tem contribuído para a crescente popularidade do Moodle, e fez dela um projeto de desenvolvimento em curso com o apoio de desenvolvedores ao redor do mundo. Porque o Moodle usa codificação em PHP, a comunidade de desenvolvedores é bastante grande e até mesmo pequenas organizações podem apoiá-lo por encontrar pessoal com experiência em programação orientada a objetos simples.

Este curso irá guiá-lo com os princípios introdutórios de criar um “bloco” que pode ser adicionado à sua instalação Moodle própria, ou – quando compartilhada – os outros sites ‘Moodle em todo o mundo, como um acessório opcional.

Desenvolvedores Moodle fizeram progressos significativos e impressionantes para fazer Moodle um sistema de gestão de aprendizagem que é amplamente adotado em todo o mundo. Uma das áreas críticas onde se comparado favoravelmente às alternativas comerciais é a sua atenção para a acessibilidade universal. Enquanto os fornecedores comerciais têm sido lentas para responder às

preocupações levantadas pelos usuários de acessibilidade, o ambiente de código aberto do Moodle para as adaptações a serem feitas mais rapidamente. Respeito pelos princípios de “acessibilidade universal” garante que todos os usuários, independentemente da sua entrada de computador e dispositivos de saída, têm acesso equivalente. Indivíduos com deficiência física são particularmente afetados quando usam tecnologias assistivas para acessar a World Wide Web. É imperativo que os programadores de aderir aos princípios de acessibilidade como eles criam melhorias e modificações Moodle para seus usuários ou a comunidade Moodle em geral para que a adoção do Moodle e utilização não cair devido ao fato de não cumprir normas de acessibilidade. Portanto, na lição desta semana, você encontrará uma introdução à acessibilidade web que fornecem a base para as referências à acessibilidade que ocorrem no decorrer deste curso.

A comunidade Moodle reúne para troca de informações em Moodle.org. Esta primeira semana, como parte de nossas atividades introdutórias, crie uma conta para si mesmo em Moodle.org e login para explorar por si mesmo a riqueza de informações lá.

Use esta semana para também rever os fundamentos de SQL e PHP para preparar-se para as atividades nas próximas semanas.

Esperar para ver cada unidade deste curso no mesmo formato:

\* Visão Geral

\* “Fazer” lista resumindo todas as atividades dos participantes para a semana, incluindo sites de tarefas de leitura ou de suporte web para visitar

Confira a programação do curso e bloco Calendário curso para prazos de atribuições.

Certifique-se de postar qualquer dúvida que você tenha no “Perguntas?” fórum na seção superior da página principal deste curso.

“To Do” Lista.

Novas janelas do navegador abertas para cada recurso externo.

1. Apresente-se aos participantes da classe outras pessoas no Fórum de discussão designado.

Cumprimente os outros.

2. Visite Moodle.org. Participe da comunidade on-line, criando uma nova conta. Navegue pelo site, incluindo a seção de Desenvolvimento Moodle e os links no menu principal na margem esquerda de “Documentação”, “Moodle Buzz,” e “Issue Tracker”. Dê uma olhada no Tracker e localizar uma passagem em aberto que lhe interessa.

3. Enviar a URL do bilhete aberto selecionado no Bug Tracker no “Open Ticket” área de atribuição deste curso.

4. O que exatamente é “open source” software? Leia a definição de Open Source (Anotada) da Open Source Initiative.

5. Leia Introdução à Acessibilidade Web do W3C Web Accessibility Initiative. Certifique-se de explorar as informações relacionadas abrangente sobre ... Deficiência diferentes que podem afetar a Acessibilidade Web.

6. Analisar cuidadosamente a lista de verificação Seção WebAIM 508. Estes padrões são amplamente adotado e melhorar significativamente a acessibilidade. O que é “Seção 508?”

“Seção 508 (a Emenda de 1998 a Seção 508 do [dos EUA] Rehabilitation Act) exige que a [EUA] governo federal para fazer todos os bens e serviços, incluindo páginas Web, totalmente acessível. Identifica normas específicas para Internet e acessibilidade Web, que são muitas vezes utilizados como base para avaliar se ou não sites atender requisitos de acessibilidade.” University of Maryland University College de Acessibilidade na Educação a Distância Glossário.

7. Revisão esta descrição de Programação orientada a objetos da Wikipedia.

8. Leia o Moodle seis primeiros Coding Guidelines, bem como a seção Estilo de codificação.

9. Completar os tutoriais para o PHP (PHP Tutorial – de Dados Refsnes) e SQL (SQL Curso – Interactive Training Online do SQL para iniciantes). Se você já estiver familiarizados com os conceitos básicos de PHP e SQL, mas pode usar uma revisão para refrescar sua memória prática,

com estes tutoriais online. Foco na lista PHP Básico para entender completamente, assim como Incluir, Sessions, e data a partir da lista PHP Avançado. Olhar através de outros tópicos PHP Avançado também, como e-mail e de arquivos.

10.Rever a terminologia importantes incluídos no glossário, bem como os tutoriais PHP e SQL antes de fazer o quiz.

11. Faça o teste antes do prazo anotado no calendário do curso (canto superior esquerdo da página do curso principal). Duas tentativas são permitidas.

12. Coloque sua resposta para as questões para discussão atribuído logo após o início da Unidade 1 quanto possível. Preste atenção para as respostas dos outros e responder a pelo menos dois colegas de classe – começando com aqueles sem respostas já.

## Unidade 2: Criando e trabalhando no ambiente de desenvolvimento

### Visão geral | Unidade 2 “fazer” lista

Recursos externos abertos em novas janelas do navegador.

Nesta unidade, estaremos examinando uma variedade de ferramentas para fazer o trabalho de criação e edição de código do Moodle mais fácil. Como acontece com qualquer tarefa criativa, ter as ferramentas certas fará o trabalho mais fácil. Ao fazer a mecânica mais transparente, que nos permite concentrar nossos pensamentos sobre o processo criativo (o código real que queremos) em vez de se atrapalhar com o processo de gravação de nossos pensamentos.

Existem várias ferramentas de desenvolvimento que será introduzido nesta semana junto com exercícios práticos para que possa explorar as características de cada um.

O browser dois plug-ins introduzido ajudar na criação e depuração de problemas com o navegador. Ambos são instrumentos preciosos para ajudar a solução de problemas – especialmente os problemas CSS e Javascript.

diagrama de Venn de software descritos neste unitYou pode estar familiarizado com outras ferramentas que executam tarefas similares – por favor, compartilhar essas experiências na discussão esta semana.

Mais desenvolvimento para o Moodle começa com um programador trabalhando em seu computador local. A fim de desenvolver localmente você precisa primeiro configurar o seu ambiente de desenvolvimento. Nesta unidade, vamos instalar e configurar as ferramentas que você precisa desenvolver para Moodle localmente.

Web local e servidor de banco

Moodle requer um servidor web que pode interpretar o código PHP, bem como um servidor de banco de dados para que você pode salvar o conteúdo. XAMPP é um servidor web e banco de dados pré-configurado para um conjunto específico de sistemas operacionais. XAMPP foi pré-configurado para quatro sistemas operacionais Windows, Mac OS X, Linux e Solaris. Vamos nos concentrar em Windows, Mac OS X e Linux como ambientes de desenvolvimento.

XAMPP, MAMP e LAMPP são siglas para Windows, Mac ou Linux, respectivamente, e Apache, MySQL, Perl e PHP.

### Ambiente de Desenvolvimento Integrado

Entrada revisão Wikipedia está descrevendo o ambiente de desenvolvimento integrado (IDE).

Depois de instalar um servidor web e banco de dados, você precisará instalar e configurar o seu ambiente de desenvolvimento integrado (IDE). Na Humboldt State University usamos Eclipse, que é uma fonte aberta livre baseado em Java IDE. Outras Universidades, como San Francisco State University uso PhpED, que também é uma IDE comercial caracterizado cheio com um depurador PHP. Outros ambientes de desenvolvimento potencial de desenvolvimento do Moodle são vim, da Adobe Dreamweaver, Nvu ou qualquer editor de texto. Para os fins deste curso, vamos usar o

Eclipse.

Alternativa Ambientes de Desenvolvimento

PhpED

PhpED é um IDE somente para Windows desenvolvido por NuSphere. Tem um depurador integrado e a maioria dos mesmos recursos que o Eclipse.

VIM

Vim é um editor de texto altamente configurável construído para permitir a edição de texto eficiente. É uma versão melhorada do editor vi distribuído com a maioria dos sistemas UNIX.

Vim muitas vezes é chamado de “editor do programador” e é tão útil para a programação que muitos consideram uma IDE inteira. Não é só para programadores, no entanto. Vim é perfeito para todos os tipos de edição de texto, de compor e-mail para editar arquivos de configuração. Instruções sobre como configurar o Vim para uso com o Moodle.

Adobe Dreamweaver

Dreamweaver é uma ferramenta comercial comumente utilizado para desenvolvimento web que também pode ser usado para desenvolver programas PHP. Ele vai funcionar tão bem para o desenvolvimento de Moodle, mas carece de algumas das características de uma IDE como o Eclipse total ou PhpED.

Navegador desenvolvedor Barras de Ferramentas

**Firebug para Firefox**

Firebug é um plugin para o navegador Firefox indispensável para dinamicamente a visualização e edição do HTML, Javascript e CSS código-fonte de uma página web.

Selênio para o Firefox

O selênio é uma ferramenta de teste para aplicações web. Testes Selenium executado diretamente em um navegador e agir como um usuário real faria. Um teste de selênio será executado no Internet Explorer, Mozilla e Firefox em Windows, Linux e Macintosh. Execução Selenium para outros navegadores envolve o download do núcleo Selênio e descompactação da pasta principal para o diretório raiz do seu web instalar o XAMPP.

Molibdênio para o Firefox

O molibdênio é um outro add-on para Firefox que pode editar e executar os testes com Selenium ponto de exclamação

Dica importante!

Todos os validadores de links ‘para testes executados em sites externos, de modo que o URL usado para validar os sites devem estar publicamente visível. Para utilizar os validadores em seu endereço de localhost você terá que abrir a porta 80 no firewall do Windows, que você deve ser convidado a fazer a primeira vez que você iniciar o Apache através do XAMPP. Você também vai precisar de usar o endereço IP para o seu computador em vez de localhost (por exemplo, <http://69.57.234.89/test/test.php>). Se você está atrás de um router firewall você terá que encaminhar a porta 80 para o computador que você está usando. Se você estiver usando DSL, o seu endereço IP provavelmente irá mudar numa base diária, o que significa que você vai precisar alterar localhost para o seu novo endereço IP para os validadores para o trabalho.

Web Developer para Firefox

A extensão **Web Developer** é projetado para Firefox, Flock, Mozilla e Firefox, e será executado em qualquer plataforma que suporte esses navegadores, incluindo Windows, Mac OS X e Linux. (Nota da Ajuda disponível se você precisar de ajuda com a instalação). O Web extensão desenvolvedor é uma excelente ferramenta para a visualização de formas variáveis, CSS e HTML em linha, bem como validar css, html e acessibilidade. Patrick Lauke faz um excelente trabalho de explicar como usar a extensão Web Developer para validação de acessibilidade, bem como explicando as diferentes partes do cumprimento 508 – ver a leitura atribuídas nesta Unidade “fazer” lista.

## Web Developer para o Internet Explorer

O Web Developer Toolbar para Internet Explorer é semelhante ao de extensão Web Developer para Firefox. Ele permite que você:

- \* Explorar e modificar o modelo de objeto documento (DOM) de uma página Web.
- \* Localizar e selecionar elementos específicos em uma página Web através de uma variedade de técnicas.
- \* Seletivamente desabilitar as configurações do Internet Explorer.
- \* Ver HTML objeto nomes de classes, IDs e detalhes como caminhos de ligação, os valores de índice de tabulação e teclas de acesso.
- \* Tabelas Contorno, células da tabela, imagens ou tags selecionadas.
- \* Validar HTML, CSS, WAI, e links feed RSS web.
- \* Display dimensões da imagem, tamanho do arquivo, informações sobre o caminho, e texto (ALT) alternativo.
- \* Imediatamente redimensione a janela do navegador para uma nova resolução.
- \* Seletivamente limpar o cache do navegador e cookies salvos. Escolha a partir de todos os objetos ou aqueles associados com um determinado domínio.
- \* Apresentar uma régua de design com todos os recursos para ajudar a alinhar e medir com precisão objetos em suas páginas.
- \* Encontrar as regras de estilo utilizada para definir os valores estilo específico em um elemento.
- \* Ver a fonte formatado e sintaxe colorida de HTML e CSS.

## Editor de XMLDb

XMLDb é uma camada de abstração de banco de dados que fornece uma maneira comum de criar e atualizar tabelas de banco de dados para o Moodle. Definição de cada tabela (campos, chaves e índices) é salvo em um formato XML. Este arquivo é lido quando o banco de dados do Moodle é criada pela primeira vez e sempre que um módulo ou plugin é instalado. Isso significa que os desenvolvedores a criar uma definição de tabela (s) usado pelo seu módulo ou plugin e administradores de sistema podem optar por utilizar qualquer sistema de banco de dados suportados, tais como Oracle, MSSQL, MySQL ou PostgreSQL. Você vai usar o Editor de XMLDb na Unidade 7, quando a edição de banco de dados do Moodle. Mais detalhes serão fornecidos na Unidade 7.

## Acessibilidade Web

Sites avaliar a acessibilidade é facilitada por uma variedade de verificadores de acessibilidade e ferramentas de validação. Enquanto verificações manuais são quase sempre necessárias, o software pode agilizar a avaliação de um site local e os erros que poderia ser negligenciada pelo designer numa avaliação manual. A leitura atribuído para esta unidade, Patrick Lauke de “Avaliação de Acessibilidade Web Sites com o Firefox,” ilustra como barra de ferramentas do Firefox Desenvolvedor Web podem ser valiosos na detecção de problemas de acessibilidade.

Há certamente outras ferramentas de software disponíveis para aumentar a produtividade no ambiente de desenvolvimento e verificar se há problemas de acessibilidade universal. Se você já tentou outras ferramentas que você prefere, por favor compartilhe as suas experiências no fórum desta Unidade de discussão.

topo

“To Do” Lista

1.

Instalar um servidor Web e banco de dados local  
Instale XAMPP / MAMP / LAMPP

Siga as instruções abaixo para seu sistema operacional para baixar e instalar XAMPP / MAMP / LAMPP:



Janelas <http://www.apachefriends.org/en/xampp-windows.html>  
Mac OS X <http://www.apachefriends.org/en/xampp-macosx.html>  
Linux <http://www.apachefriends.org/en/xampp-linux.html>  
Iniciar o Apache e MySQL

Para iniciar o Apache eo MySQL, localize a pasta em que você instalou o XAMPP, MAMP ou LAMPP e encontrar o ícone correspondente e dê um duplo clique nele.

### Unidade 3: gerenciamento de código fonte

#### Visão geral | Unidade 3 “fazer” lista

Recursos externos abrem em novas janelas do navegador.

Sistemas de gestão de origem são essenciais para o desenvolvimento de software.

“Não usar um é como dirigir um carro muito rápido: é divertido e você pode chegar ao seu destino mais rápido, mas um acidente é inevitável.” M. Tim Jones no controle de versão para Linux: Visão de arquiteturas, modelos e exemplos.

Sistemas de gestão fonte fornecem uma maneira confiável para:

- \* Controlar alterações
- \* Colaborar com outros desenvolvedores
- \* Fornecer gerenciamento de arquivos
- \* Reverte para versões antigas
- \* Versioning – agrupamento e tagging arquivos juntos para que você possa recuperar a configuração exata em um ponto posterior no tempo.

Nesta unidade abordaremos conceitos básicos necessários para usar o gerenciamento de código fonte para desenvolvimento de software.

Para os fins deste curso, você estará usando CVS exclusivamente, mas em seu desenvolvimento Moodle diária haverá momentos em que você estará usando SVN, CVS, ou ambos.

topo

#### “To Do” Lista

1.

Leia sobre como usar o CVS para controle de versão de código fonte

Pensamentos tutorial usando Fluid ‘CVS, ler sobre os comandos comumente usados CVS.

Certifique-se de cobrir adicionar, cometer, atualizar e remover. Embora estes comandos são semelhantes aos comandos SVN, alguns deles se comportam de maneira diferente do que suas contrapartes SVN.

2.

Leia sobre como usar o SVN para controle de versão de código fonte

Ver as opções disponíveis para a instalação de SVN. Depois de analisar algumas das opções de instalação disponíveis, verifique se você tiver um deles instalado no seu ambiente de desenvolvimento. Se você não tiver um instalado, escolha um pacote de instalação apropriado e siga as instruções para instalar esse pacote.

Leia a seção, “Checkout inicial” no capítulo 2 do livro online, “Controle de Versão com Subversion”, que descreve os passos que se aplicaria a verificar uma cópia de trabalho do Moodle a partir de um servidor SVN.

Percorrer a “Basic trabalho realizado no ciclo” no Capítulo 2 do livro online, “Controle de Versão com Subversion”, para estudar os comandos que é provável que você precisa durante um dia de

trabalho típico. Preste atenção especial aos cinco comandos que você usará mais:

- \* Svn update
- \* Svn status
- \* Svn commit
- \* Svn add
- \* Svn delete

Rever os outros comandos na seção deste capítulo para que você entenda como e quando eles são usados. Lembre-se que

svn ajudar <comando> (por exemplo, ajuda svn commit)

irá mostrar-lhe desejar que o comando é usado para e algumas das opções válidas. Se você estará usando TortoiseSVN então dê uma olhada no manual de ambos disponíveis através da aplicação ou online em TortoiseSVN.

Leia Melhores Práticas de Subversion do Open Collabnet.

3.

Obter Moodle para o seu ambiente de desenvolvimento.

Usando o CVS

CVS para Windows

1. Encontrar o diretório Moodle em C: \ XAMPP \ htdocs.
2. Direito do mouse clique sobre o diretório moodle e escolha “CVS Checkout” no menu. Você deverá ver uma caixa de diálogo.
3. Digite este texto no campo CVSROOT:

: Pserver: anonymous@SERVER.cvs.moodle.org: / cvsroot / moodle (para desenvolvedores norte-americana, substitua o “servidor” com “nós”

TortoiseCVS verificação captura de tela módulo

4. Sob o “Módulo” campo, digite “moodle” para obter o diretório moodle.
5. Clique no botão “Revisões” tab.
6. Selecione o “ramo Escolha ou tag” botão de rádio.
7. Clique no botão “Atualizar a lista ...” botão.
- 8.

ponto de exclamação

Dica importante!

Se você quiser verificar para fora uma versão diferente do Moodle (por exemplo, 1.9 ou 2.0) ou uma versão específica (por exemplo 1.8.4 ou 1.8.3) você vai querer mudar o nome no ramo ou campo de nome da marca. Depois de clicar no “Atualizar a lista? Botão” você verá uma lista no drop down de todos os ramos Moodle possível check-out.

Mais tarde, para atualizar sua cópia local do moodle para a versão mais atual 1.8 em CVS, apenas direito do mouse clique no diretório do moodle e escolha “CVS Update”.

Selecione MOODLE\_18\_STABLE da lista de todas as revisões Moodle possível.

TortoiseCVS verificação módulo captura de tela revisão

9. Pressione o botão “OK” e tudo deve ser verificado para o diretório moodle.

CVS para Mac e Linux

Verifique se você tem um cliente CVS instalado e funcionando em seu ambiente. Se você estiver trabalhando em um ambiente Linux ou Unix você já deve ter CVS instalado. Se você estiver trabalhando em um ambiente Mac OS X você também deve já tem CVS instalado que pode ser acessado através do prompt de comando. Se você estiver trabalhando em um ambiente Windows baixar CVS de TortoiseCVS.

Explorar esta documentação sobre check-out uma cópia do CVS Moodle localizado em Moodle.org

e Cartão de Referência Rápida CVS.

Para conectar e login pela primeira vez para o servidor CVS:

```
cvs-d: pserver: anonymous@moodlecvs.sfsu.edu: / cvsroot / moodle de login
```

Não há senha, quando pediu para um, apenas pressione Enter. Para fazer o checkout (download) o código do Moodle toda a pela primeira vez, use este comando para obter a última versão STABLE:

```
cvs-z3-d: pserver: anonymous@moodlecvs.sfsu.edu: / cvsroot / moodle co-r  
MOODLE_18_STABLE moodle
```

Ou a última versão de desenvolvimento:

```
cvs-z3-d: pserver: anonymous@moodlecvs.sfsu.edu: / cvsroot / moodle moodle co
```

topo

Mais tarde, para atualizar sua cópia local do Moodle para a versão atual do CVS, você só precisa entrar em seu diretório Moodle local e digite:

```
cvs update-dP
```

Para atualizar sua cópia local e para salvar o log do processo:

```
cvs update-dP-r MOODLE_18_STABLE | tee upgrade.log
```

Para atualizar sua cópia local do Moodle para uma nova versão (por exemplo, de 1,7 + 1,8), vá para o diretório Moodle local e digite:

```
cvs update-dP-r MOODLE_18_STABLE
```

Então olha para o upgrade.log, nomeadamente procurar linhas que começam com “C” (conflito):

```
‘^ C’ grep upgrade.log
```

trickNote mágica!

Conflitos podem aparecer se você tiver modificado manualmente os arquivos de origem. Você tem que resolver os conflitos antes de usar o site. Veja CVS para desenvolvedores para obter mais detalhes.

Outras alternativas para obter o código Moodle

Windows

Download de “Pacotes de Moodle para o Windows” do Moodle.org. Escolha o Moodle 1.8.x versão + (para Windows, geralmente é chamado como “MoodleWindowsInstaller-latest-18.zip ‘).

Mac OS X

Você vai encontrar algumas informações sobre CVS e Mac OS X na documentação para a instalação do pacote completo Moodle4Mac. Por favor, leia Como atualizar seu Moodle4Mac. Ele funciona bem com os novos servidores CVS.

Use as informações desta Unidade para selecionar o tipo de caixa que se aplica a você, em seguida, check-out uma cópia do Moodle para um diretório no seu webroot local do seu ambiente de desenvolvimento.

4.

Faça o teste.

5.

Contribuir para o tópico de discussão.

Qual o método que você usou para fazer checkout Moodle para o seu ambiente de desenvolvimento? Quais são as circunstâncias que influenciaram a sua decisão de usar o SVN ou CVS?

SVN (Subversion) ou CVS (Concurrent Versioning System) para o Desenvolvimento Moodle?

## Apenas usando CVS

A configuração mais comum para desenvolvedores de versões de código fonte com Moodle é usar CVS e check-out a versão mais recente do ramo Moodle estável do repositório CVS do Moodle. Nesta configuração, o desenvolvedor obtém todas as atualizações através do Moodle Moodle repositório do CVS. O desenvolvedor então faz alterações localmente e testes com a versão atual do CVS Moodle. O desenvolvedor também pode decidir fazer o checkout da CVS contribuir ramo do repositório Moodle, a fim de testar e modificar terceiros Módulos e Plugins. O desenvolvedor pode ou não ter permissões para confirmar as alterações de volta ao repositório CVS do Moodle ou sucursais. Neste cenário os comandos mais comuns o desenvolvedor precisa saber são CVS update e, se eles têm privilégios cometer, CVS cometer.

Só quem faz CVS trabalha?

- \* Os desenvolvedores que só quer ajudar Moodle
- \* Os desenvolvedores que não têm um ambiente de produção
- \* Os desenvolvedores que se sentem confortáveis em desenvolvimento em seu ambiente de produção
- \* Desenvolvedor grupos que não precisam compartilhar seus códigos com outras pessoas em seu grupo
- \* Os desenvolvedores que querem controle de versão de seu código fonte próprio, mas não me importo sobre o controle de versão do Moodle

topo

## Apenas usando SVN

Esta opção é para pessoas que preferem fazer o download do Moodle em um formato zip e precisam de uma forma de controle de versão do código que estão criando. Pode haver várias configurações para isso, com o mais provável que você baixar o código fonte a partir de Moodle <http://www.moodle.org> e mesclá-lo em seu repositório SVN. Você gerenciar todo o seu código através de SVN. Neste caso, você poderia usar todos os comandos dentro SVN, mas os comandos mais provável será svn update, svn commit e svn merge.

Só quem faz SVN trabalha?

- \* Alguém que gostaria de controle de versão o seu próprio código, mas não se preocupam com o controle de versão para o Moodle
- \* Um grupo de desenvolvedores que estão em locais diferentes mas que partilham uma base de código comum

## Usando CVS e SVN Juntos

Esta opção é a mais comum para uma organização que tem vários desenvolvedores e vários servidores. Os desenvolvedores geralmente o checkout da versão mais atual do Moodle a partir do repositório CVS e depois mesclar as alterações em seu repositório SVN. Isso permite que eles usem todo o poder do SVN, minimizando o tempo que leva para obter o bug mais atual e correções de segurança dos desenvolvedores moodle.org 's. Neste cenário você irá usar o comando update CVS junto com potencialmente todos os comandos SVN.

Quem faz o CVS e SVN trabalha?

- \* Vários grupos de desenvolvedores
- \* As organizações com vários servidores que usam diferentes módulos ou blocos
- \* Único desenvolvedores que estão interessados em aprender SVN

Se você vai se desenvolver no Moodle sem utilizar SVN para controle de versão, então você só precisa seguir os passos para verificar para fora do Moodle, de um repositório CVS.

Se você vai ser em desenvolvimento em Moodle em uma loja que gerencia uma base de código no SVN, então você pode precisar apenas de seguir os passos para verificar para fora do Moodle, de um repositório SVN.

Em alguns casos, como em Courseware Humboldt State University do Centro de Desenvolvimento,

um repositório SVN usado pela maioria dos desenvolvedores é mantido atualizado através de atualizações de um repositório CVS Moodle. Neste caso, o desenvolvedor administrador / chumbo deve ser entendido com ambos os sistemas. Se este for o seu caso, siga as etapas para a verificação de cópias de ambos os sistemas.

## Unidade 5 Parte A: Estrutura de Diretórios do Moodle

### Visão geral | Unidade 5 “fazer” lista

Recursos externos abertos em novas janelas do navegador.

Esta Unidade se refere à estrutura de diretórios no Moodle versão 1.8.

O bloco de administração / admin directorysite

Arquivos sob este diretório lidar com todos os recursos sob o “Site Administration” bloco.

Em novembro de 2007, há 635 arquivos espalhados em 137 diretórios sob o diretório / admin (no Moodle 1.8.3 + versão). Para os propósitos deste curso, vamos concentrar apenas no arquivo mais importante: / admin / cron.php

Alguns dos módulos do Moodle precisam de verificação contínua para executar tarefas. Por exemplo, Moodle precisa verificar os fóruns de discussão para que ele possa enviar as mensagens para as pessoas que inscreveram. O script que executa essas verificações contínua está localizado no diretório admin e é chamado cron.php. Geralmente montamos um mecanismo para esse script é executado regularmente, por si só, embora você também pode executá-lo manualmente diretamente de um navegador da Web, ou a partir de uma linha de comando. Cron.php cria “batimento cardíaco” para desempenhar funções em intervalos definidos por cada módulo. Este tipo de mecanismo é tipicamente implementado usando um daemon cron / serviço através do cron.

O script cron.php:

1. Procura na tabela mdl\_modules (assumindo que o prefixo da tabela padrão, é claro) no banco de dados para os módulos Moodle programado para ter as suas funções de executar cron
2. Em cada módulo, tais localiza diretório qualquer função chamada módulo name\_cron-in o arquivo e executa-lo lib.php
3. Procura na tabela mdl\_block para os blocos programados para os seus métodos de cron (funções de objeto) para ser executado
4. para cada bloco tal, executa o método cron para um novo objeto associados a esse bloco

Esses arquivos (os arquivos lib.php e os arquivos de onde as classes são definidas bloco) pode conter funções de limpeza, funções de e-mail ou qualquer coisa que precisa ser executado em uma base regular. Por exemplo, o cron irá acionar o sistema para criar os backups dos cursos no horário especificado nas definições de administração. Também desencadeia todas as notificações de e-mail mensagens módulo ou forum, mas nem todas as funções são chamadas cada vez que o cron será executado. Algumas funções, como desinscrever estudantes que não têm logado ou excluir cópias antigas dos arquivos de log, só são executadas ocasionalmente. O arquivo cron.php tem uma seção que irá chamar aleatoriamente essas tarefas centrais aproximadamente 1 em cada 5 vezes o cron será executado.

Note-se que a máquina executando o cron não precisa ser a mesma máquina que está executando o Moodle. Por exemplo, se você tem um serviço de hospedagem limitado, que não tem um serviço de cron, então você pode optar por executar cron em outro servidor ou no computador de casa. Tudo o que importa é que o arquivo cron.php seja chamado regularmente.

O carregamento desse script não é muito alta, dessa forma 5 minutos usualmente é razoável, mas você pode reduzir o período de tempo para 15 minutos ou até mesmo 30 minutos. É melhor não fazer o período de tempo demasiado longo, como atrasar envio de emails podem diminuir a atividade em curso. Lembre-se que o correio-outs também esperar que o tempo de edição para

expirar antes de ser enfileirado para envio. Enquanto cron é executado regularmente e com bastante frequência, a carga não é muito alto. No entanto, se não tiver sido executado em um longo tempo ou a frequência processing está muito longe das cargas pode ser considerável dependendo da atividade do site e todos os processos de fundo de lote, como se matricular ou matricular-un que podem precisar de ocorrer.

topo

Os blocos / diretório

Até agora você provavelmente já sabe sobre “blocos” no moodle. Blocos Moodle tipicamente aparecem nas margens esquerda e direita da página principal de um curso. Eles geralmente contêm uma função e diferem das “seções” que aparecem abaixo a coluna do centro da página principal de um curso que contêm o conteúdo e actividades ou recursos adicionados pelo instrutor.

blocos identificados em um curso

Quando um curso Moodle é criado ele automaticamente gera vários “blocos” que lhe permitem controlar a forma como você navegar ou encontrar informações no Moodle (exemplos destes incluem o “Site Administration” bloco para o administrador, o “Fórum de Pesquisa” bloco ou o Calendário ” “bloco para todos os usuários, etc.) Além dos blocos que aparecem automaticamente, você vai notar que você tem várias opções para os blocos ainda de ferramentas (do Blocks menu drop-down, quando a edição é ligado) para adicionar ao seu curso. Exemplos incluem o “Remote RSS Feeds” bloco, o “Blog” do bloco, etc.) Da mesma forma, você pode excluir qualquer um dos blocos existentes, se você sente que são desnecessários ao seu curso e você também pode mover os blocos para que eles apareçam em um único lado, em vez de em ambos os lados do conteúdo do meio (isto só pode ser conseguido usando a mover ou excluir botões com a edição ligado).

Em um típico Moodle 1.8 instalação, você verá um conjunto de blocos padrão disponíveis para cada criador curso.

blocos de lista do menu suspenso seleção

Se você olhar para a sua instalação Moodle local e navegue até o diretório / blocos, você encontrará um diretório correspondente para cada bloco.

listagem da pasta blocos

Se você precisa depurar o “Blog Menu” do bloco, por exemplo, você ir para o “blog\_menu diretório ‘. Se você precisa depurar o” Search Forums “do bloco, começar com o” search\_forums diretório “, etc Mais tarde neste curso vamos introduzir mais sobre como escrever um bloco Moodle do ponto de vista de um programador. Detalhes adicionais sobre “administração Blocks” (geralmente a partir da perspectiva do administrador de sistemas).

truque de mágica

Dica importante:

Seqüências localizadas deve ser em um diretório do idioma separado, como **en\_custom**. O padrão é uma string, se não for encontrado em uma linguagem ele volta para o **en\_utf8**. Desta forma, as alterações substituir o padrão, mas você não tem que incluir todos os string ou gerenciar mudanças strings linguagem central. Você apenas tem que gerenciar as configurações localizadas.

O / lang

Este diretório contém todos os arquivos de idiomas diferentes, incluindo o seu conteúdo relacionado ajudar botão. O para pagar mais atenção é o diretório “en\_utf8”. Sob este diretório, você será capaz de compreender como Moodle define todas as cordas em diferentes módulos ou blocos. Por exemplo, todas as cordas que estão relacionados com o módulo de administração definido na admin.php.

O diretório / lib

Para um desenvolvedor Moodle, este é provavelmente um dos diretórios mais importantes que você

irá usar. O diretório / lib contém muitos arquivos de biblioteca (275 diretórios e arquivos de 2179, a partir de Moodle 1.8.3 +) variando de ajax e bibliotecas adodb, a bibliotecas (Yahoo User Interface) e XMLDb yui. Os três mais importantes arquivos da biblioteca são, provavelmente, **moodlelib.php**, **weblib.php** e **accesslib.php**. Estes três arquivos de biblioteca será discutido com mais profundidade na Parte B desta Unidade. No entanto, este é um breve resumo destes três arquivos de biblioteca:

moodlelib.php é o Moodle biblioteca “principal”. Este arquivo contém diversas funções de uso geral Moodle (como set\_user\_preferences (), get\_user\_timezone (), e assim por diante).

weblib.php contém funções que ajudam a produzir uma saída Moodle web. Por exemplo, tem funções como format\_text\_email (), print\_header (), print\_group\_menu (), etc

datalib.php contém funções que allow moodle para acessar o banco de dados, ele também tem funções que lidam com as capacidades de papel, como load\_defaultuser\_role (), get\_guest\_role (), etc

O diretório / mod

Este diretório contém a maioria dos principais módulos do Moodle. Atualmente em 1,8 moodle existem 18 módulos-chave no diretório / mod.

listagem do diretório mod

topo

“To Do” Lista

1.

Experiência com o script cron executando-o diretamente do seu navegador (<http://localhost/moodle/admin/cron.php>). Se cron é chamado na linha de comando por qualquer usuário conectado à sua Moodle que irá criar um ambiente de administração temporária, a fim de executar e, em seguida, registrar o usuário para fora. Você pode desativar linha de comando em execução do cron, desativando a seção apropriada no arquivo cron.php. Veja mais detalhes sobre como configurar a execução automática do script (cron.php) em (1) Windows (2), serviços de hospedagem, ou (3) Usando o programa crontab em Unix. Normalmente é esse o trabalho de um administrador de sistemas para configurar isso, é além do escopo do curso introdutório deste programador Moodle é.

2.

Executar admin / auth.php do seu navegador Web

(<http://localhost/moodle18/moodle/admin/auth.php>) para ver o que ele faz. Você também pode acessar o arquivo através do “Site Administration” menu (em Usuários – Autenticação>).

## Unidade 5 Parte B: Variáveis Globais

Recursos externos abertos em novas janelas do navegador.

Na parte B, vamos focar as variáveis globais do Moodle. Mais tarde, na Parte C desta unidade vamos introduzir Moodle API e alguns dos arquivos de biblioteca mais importante Moodle.

Globals Moodle

A maioria dos programadores PHP deve saber que, em geral, que pretende evitar o uso de variáveis globais, tanto quanto possível. Independentemente deste Moodle orientação geral ainda usa algumas variáveis globais. Como um programador Moodle, você não deve usar quaisquer outras variáveis globais que não os listados nesta lição. Em geral, quando a programação com Moodle você não deve usar built-in PHP superglobals tanto – em vez disso, você deve acessar os dados destas variáveis através de funções do Moodle API.

## \$ CFG

O objeto \$ CFG tem muitas das directivas Moodle de configuração. Em geral, quando ligados a itens que você deseja utilizar as variáveis em \$ CFG para que seu código é portátil para outros sites, e a ordem de chamada dos scripts não é crítica.

Você pode encontrar a maioria das variáveis \$ CFG no arquivo / config.php. Nós não vamos cobrir todos os atributos de US \$ CFG, mas vamos concentrar-nos sobre aqueles que são mais propensos a usar durante a programação.

- \* \$ CFG-> wwwroot: Este é o endereço base da instalação do moodle. Use este URL ao construir, de modo que não importa o que outro script inclui o seu script. Isto é, desde o seu script pode potencialmente ser incluídos a partir de um diretório qualquer que você não vai ser capaz de forma confiável para construir URLs relativos. Dê uma olhada no seu config.php e ver como esta variável foi definida.
- \* \$ CFG-> dirroot: Este é o caminho do arquivo de base da instalação Moodle. Este é o diretório onde o Moodle está instalado. Então você vai encontrar a sua blocos em “{\$ CFG-> dirroot} / blocos”
- \* \$ CFG-> dataroot: Este é o diretório de dados para o seu site Moodle.
- \* \$ CFG-> admin: Este é o nome de usuário administrador.
- \* \$ CFG-> httpswwwroot: https Se estiver habilitado este é o endereço base segura wwwroot. Se https não está habilitado este é o mesmo que \$ CFG-> wwwroot.
- \* \$ CFG-> libdir: Este é o local dos arquivos de biblioteca moodle.
- \* \$ CFG-> textfilters: É uma lista separada por vírgula dos filtros de texto atualmente habilitado.
- \* CFG-> \$ loginhttps: É um valor booleano que indica se ou não está habilitado para https login.
- \* \$ CFG-> noreplyaddress: O endereço de email para usar e-mail que não devem receber uma resposta.
- \* \$ CFG-> sesskey: O sesskey atual. Use o sesskey () para obter a chave de sessão atual usuários para o formulário e processamento de dados. Usando o sesskey () função e não o valor em \$ USER-> sesskey ou US \$ CFG-> sesskey é o preferido como o sesskey () função permite que o API para mudar e você não precisará atualizar seu código.
- \* \$ CFG-> themedir: Esta é a localização do diretório do tema através do sistema de arquivos.
- \* \$ CFG-> themewww: A web URL base acessível para o diretório do tema.
- \* \$ CFG-> Tema: É o tema do site atual.
- \* \$ CFG-> pixpath: A web acessíveis URL base do diretório pix. Este é o lugar onde os ícones moodle virá.
- \* \$ CFG-> modpixpath: A web URL base acessível dos ícones módulo.
- \* \$ CFG-> mailsessiontime: Esta variável armazena o número de sessões de hora e-mail persistir (o padrão é 1 mês). A função do cron (função de limpeza), escrito em admin / cron.php é usado para podar a tabela de sessão.

Além disso, se o seu código armazena todas as informações na tabela de configuração (geralmente nomeado como mdl\_config) depois que a informação será acessível através da variável \$ CFG também.

topo

## \$ USER

O objeto \$ USER contém informações referentes ao usuário conectado no momento. Esta variável global foi declarado em / lib / setup.php. We não irá cobrir todos os atributos deste objeto, mas em vez disso, concentrar-se nos atributos que são mais propensos a usar. A maioria desses atributos vêm do perfil de usuário.

- \* \$ USER-> id: id do usuário do Moodle.
- \* \$ USER-> auth: O método de autenticação do usuário.
- \* \$ USER-> username: username do usuário



- \* \$ USER-> idnumber: idnumber do usuário, esta é normalmente usado para identificar o usuário em um sistema externo.
- \* \$ User-> nome próprio: firstname do usuário.
- \* \$ USER-> apelido: sobrenome do usuário
- \* \$ USER-> email: O endereço do usuário de e-mail.
- \* \$ USER-> emailstop: Indica se o usuário tiver desabilitado o seu e-mail.
- \* \$ USER-> instituição: A instituição do usuário.
- \* \$ USER-> Departamento: Departamento do usuário.
- \* \$ USER-> endereço: O endereço do usuário.
- \* \$ USER-> city: cidade do usuário.
- \* \$ USER-> país: o país do Usuário
- \* \$ User-> lang: O idioma padrão do usuário
- \* \$ USER-> Tema: O tema do usuário se temas usuário é permitido.
- \* \$ USER-> lastlogin: A última vez que o usuário fez o
- \* \$ USER-> currentlogin: O tempo de login do usuário atual.
- \* \$ USER-> lastIP: A última IP do usuário logado partir.
- \* \$ USER-> imagem: Se o usuário tem um conjunto de imagem ou não.
- \* \$ USER-> timemodified: A última vez que o perfil do usuário foi modificado.
- \* \$ USER-> aluno: uma matriz de números curso id do usuário é um estudante dentro
- \* \$ USER-> timeaccess: uma matriz chave por id curso que relaciona o tempo de acesso para o referido curso particular.
- \* \$ USER-> professor: uma matriz de números de ID de usuário é claro, a um professor para.
- \* \$ USER-> teacheredit: uma matriz de números curso id do usuário é uma edição para ensinar.
- \* \$ USER-> sesskey: chave de sessão do usuário atual.

topo

## \$ CURSO

O objeto \$ CURSO contém informações referentes ao curso atualmente visualizando. Esta variável global também foi declarado em / lib / setup.php. We não irá cobrir todos os atributos deste objeto, mas em vez disso, concentrar-se nos atributos que são mais propensos a usar no futuro.

- \* \$ COURSE-> id: id do curso curso Moodle.
- \* \$ COURSE-> categoria: categoria do curso. Normalmente administrador moodle pode ter configuração de várias categorias de curso, por exemplo, “Ciência”, “Ciências Humanas”, “Saúde Pública”, etc (ou com base no semestre). Quando você cria um novo curso manualmente, por padrão, a categoria será definido como “Diversos” para a qual o ID da categoria é 1.
- \* \$ COURSE-> fullname: nome completo do curso, tais como “Computer Logic Programming Ciência Básica”
- \* \$ COURSE-> shortname: O curso abreviado, como CS101.
- \* \$ COURSE-> formato: O “formato” do curso é geralmente “site” (para o curso de raiz Moodle), “semanas”, ou “temas” (baseado no formato que o professor escolhe).
- \* \$ COURSE-> visível: Isto especifica se o curso é visível para os alunos ou não
- \* \$ COURSE-> groupmode: Isto especifica se o curso utiliza o modo de grupo ou não. 0 é não, e 1 é sim.
- \* \$ COURSE-> timecreated: O tempo que o curso foi criado. Este formato usa Unix carimbo de tempo para realmente entender o que a data real é, você precisa de alguma conversão (veja o link abaixo para mais detalhes)
- \* \$ COURSE-> timemodified: O tempo que o curso foi modificada pela última vez. Mais uma vez ele está usando Unix formato de carimbo de tempo.
- \* \$ COURSE-> convidado: “. Acesso de visitante” Se o curso permite que Se o valor for definido para 0, então não, mas se o valor for definido como 1, então sim.
- \* \$ COURSE-> startdate: A data de início do curso Moodle. Note-se que o Moodle utiliza “timestamp Unix” para converter a data real / hora para startdate. Assim, por exemplo, se você

quiser fazer uma consulta SQL para listar todos os cursos que tem data de início de 2007/08/01 a 2007/09/01 você pode escrever a consulta como esta usando “unix\_timestamp”:

selecionar fullname, nome abreviado, cat.name de mdl\_course, mdl\_course\_categories como gato onde

startdate unix\_timestamp > ( '20070801 00:00:00 ' ) e startdate <unix\_timestamp ( '20070901 00:00:00' )

e categoria order = cat.id por cat.name

Mais informações sobre o timestamp Unix.

Para outros atributos no objeto \$ COURSE, navegar na tabela mdl\_course Moodle.

Name	Default Value	Type
\$COURSE		Object(stdClass) (44 elements)
id	“1”	String
category	“0”	String
sortorder	“1000”	String
password	“”	String
fullname	“Moodle 1.8.x +”	String
shortname	“mdl18x”	String
idnumber	“”	String
summary	“test”	String
format	“site”	String
showgrades	“1”	String
modinfo	NULL	Undefined
newsitems	“3”	String
teacher	“Teacher”	String
teachers	“Teachers”	String
student	“Student”	String
students	“Students”	String
Guest	“0”	String
Startdate	“0”	String
Enrolperiod	“0”	String
Numsections	“0”	String
Marker	“0”	String
Maxbytes	“0”	String
Showreports	“0”	String
Visible	“1”	String
Hiddensections	“0”	String
Groupmode	“0”	String
Groupmodeforce	’0’	String
Lang	“”	String
Theme	“”	String
Cost	“”	String
Currency	“USD”	String

Timecreated	"0"	String
Timemodified	"1197679457"	String
Metacourse	"0"	String
Requested	"0"	String
Restrictmodules	"0"	String
Expirynotify	"0"	String
Expirythreshold	"0"	String
Notifystudents	"0"	String
Enrollable	"1"	String
Enrolstartdate	"0"	String
Enrolenddate	"0"	String
Enroll	""	String
Defaultrole	"0"	String

[top](#)

topo

\$ SITE

Variáveis site são configurações que determinam como suas funções site inteiro Moodle. As variáveis \$ site são as configurações mais técnicas. Esta é a única variável global que não é declarada no lib // arquivo setup.php. Estes são alguns atributos comumente usados em Moodle 1.8:

\* \$ SITE-> fullname: nome completo O site do Moodle. Este é baseado no que você descreveu durante o processo de instalação do Moodle.

\* \$ SITE-> shortname: nome curto O site do Moodle. Esta é também baseado no que você é chamado durante o processo de instalação do Moodle.

\* \$ SITE-> numsections: O número site Moodle de seções.

\* \$ SITE-> id: O Moodle número de identificação site. Por padrão, ele é designado como "1".

\* \$ SITE-> categoria: O Moodle número da categoria site. Por padrão, ele é designado como "0", que é a categoria raiz. Geralmente apenas o Moodle "curso de raiz" – a primeira página do Moodle, que é atribuída a categoria 0. A categoria de curso "Diversos" é categoria 1.

\* \$ SITE-> senha: Esta é a senha stite admin admin, que é o administrador super (nome de usuário: admin) senha. Nota: Não confunda este com os US \$ dbpass CFG-> – estes dois são diferentes. Os US \$ CFG-> dbpass é geralmente definida no config.php e é a senha do banco de dados do Moodle.

Name	Default Value	Type
\$SITE		Object(stdClass) (44 elements)
Visible	"1"	String
Timemodified	"1182375379"	String
Timecreated	"0"	String
Theme	""	String
Teachers	"teachers"	String
Teacher	"teacher"	String
Summary	"test"	String
Students	"students"	String
Student	"student"	String
Startdate	"0"	String
Sortorder	"1000"	String

Showreports	“0”	String
Showgrades	“1”	String
Shortname	“moodle18p”	String
Restrictmodules	“0”	String
Requested	“0”	String
Password	“”	String
Numsections	’0’	String
Notifystudents	“0”	String
Newsitems	“3”	String
Modinfo	NULL	Undefined
Metacourse	“0”	String
Maxbytes	“0”	String
Marker	’0’	String
Lang	“”	String
Idnumber	“”	String
Id	“1”	String
Hiddensections	“0”	String
Guest	“0”	String
Groupmodeforce	“0”	String
Groupmode	“0”	String
Fullname	“Moodle 1.8.x+”	String
Format	“site”	String
Expirythreshold	“0”	String
Expirynotify	“0”	String
Enrolstartdate	“0”	String
Enrolperiod	“0”	String
Enrolenddate	“0”	String
Enroll	“”	String
Defaultrole	“0”	String
Currency	“USD”	String
Cost	“”	String
Category	“0”	String

**Nota: Outras variáveis globais, como \$ THEME, \$ db, \$ MCACHE, e US \$ HTTPSPAGEREQUIRED são encontrados no / lib / setup.php.**

“To Do” Lista

1.

Leia Diretrizes Moodle adicionais de codificação: # 10 na seção de regras gerais; # 6 na seção questões de segurança.

2.

Use print\_object () (definida em / lib / datalib.php em torno da linha 1610), print () (<http://us2.php.net/print>) ou echo () ([http://us2.php.net / manual / en / function.echo.php](http://us2.php.net/manual/en/function.echo.php)) para escrever alguns dos valores das variáveis globais mencionados nesta unidade para fins de depuração.

3.

Contribuir com suas idéias às discussões para esta Unidade de variáveis globais.

## Unidade C Parte 5: Bibliotecas Moodle

### Visão geral | Unidade 5 Parte C “fazer” lista

Recursos externos abrem em novas janelas do navegador.

truque de mágica

Dica importante:

Discutir com a equipe principal Moodle (usando moodle.org “Developer Geral” do fórum) no planejamento de mudanças a qualquer um destes três arquivos, e de preferência (se você tiver acesso de gravação à árvore CVS do Moodle) depois de aprovado pela equipe principal do Moodle, você deve confirmar as alterações para o CVS Moodle

Na parte A desta unidade você foi apresentado a um dos diretórios mais importantes para os desenvolvedores Moodle, o diretório / lib. Na parte C, você vai aprender mais sobre esse diretório, especificamente os três grandes arquivos de biblioteca: moodlelib.php, weblib.php e datalib.php.

Ao revisar exemplos de código no Moodle e começar a preparar o código como parte desta lição, consulte a seção “Programação Geral e Visão geral de segurança” Unit neste curso.

Para uma visão geral, levará algum tempo para percorrer o Core Moodle API, geralmente coberto por esses três arquivos, para que você possa ter uma idéia da escala desses três arquivos de biblioteca. Esta página é gerada automaticamente usando PHPedit (uma ferramenta de software PHP) dos três arquivos de biblioteca principal no moodle / lib.

Programadores Moodle muito poucos poderia dominar cada função ou classe entre esses três arquivos de biblioteca. Normalmente você olhar para cima destes três arquivos de biblioteca importante para o que você precisa saber quando você está solucionando um problema. Existe um site muito útil fornecida pelo Moodle, <http://xref.moodle.org/> (referida neste curso como “xref”). Navegue neste site para obter idéias de como usar este site para procurar funções específicas, classes, ou variáveis.

Pois pode não ser fácil de dominar cada função dentro destes três arquivos de biblioteca, é provavelmente melhor para explicar com alguns exemplos. Aqui estão três exemplos de como você pode solucionar problemas com esses três arquivos.

#### Caso A – moodlelib.php

Como já aprendemos com a Parte A, o moodlelib.php É o arquivo de biblioteca principal de diversas funções de propósito geral Moodle. Isso significa que ele também contém funções relacionadas com login, etc Se olharmos para um caso exemplo MDL-10375 no Bug Tracker você vai notar que o problema era mnethostid indefinido (que você pode imaginar que significa “id anfitrião Mnet”). Na descrição do relatório de bug que menciona que o plano é criar uma nova create\_guest\_user () funcionam sob moodlelib.php com as configurações corretas. Este é o tipo de função que normalmente vai para moodlelib.php

#### Caso B – weblib.php

Esta é a biblioteca de todas as funções de propósito geral Moodle PHP e constantes que produzem saída HTML. No código do Moodle 1.8, que define uma classe (tabobject) e nove funções. Veja um caso de exemplo usando weblib.php com MDL-11506 no Bug Tracker.

#### Caso C – datalib.php

Esta é a biblioteca que contém funções para manipulação de banco de dados. No código do Moodle 1.8, datalib.php define 43 funções. Alguns exemplos são get\_admin () (retorna \$ objeto de usuário do usuário administrador principal), **search\_users () (pesquisa através de usuários é claro),**

**get\_users () (retorna um subconjunto de usuários), get\_courses () (retorna lista de cursos, por todo site, ou categoria), get\_logs (), etc**

O datalib.php às vezes pode ser confundida com dmllib.php, que é outra biblioteca muito importante que contém funções para manipulação de banco de dados também. No entanto, dmllib.php em negócios em geral com o nível recorde (nível mais micro) tipo de manipulação de banco de dados, enquanto lida datalib.php com os usuários, administradores, cursos, logs e outros níveis mais elevados de manipulações de banco de dados. Veja dois exemplos usando datalib.php no Bug Tracker: MDL-11135 e MDL-8549.

Durante upgrades de versão principal do Moodle, geralmente haverá mais funções adicionais que esses três arquivos (moodlelib.php, weblib.php e datalib.php).

Outras bibliotecas importantes  
ponto de exclamação

Lembre-se:

Assumir uma função existe antes de escrever a funcionalidade em seu próprio país.

API Moodle é bastante robusto, e funções podem ser divididas em várias categorias principais. Compreender onde partes do API estão localizados na estrutura do Moodle lib pode ser extremamente útil para determinar onde para verificar se uma determinada função existe ou como chamar uma função particular.

Existem algumas funções que são usados com frequência com a qual você deve se familiarizar. Como você ficar mais familiarizado com Moodle através da experiência de programação torna-se óbvio que estas são funções. Vamos esperar até começar a escrever algum código, antes de cobrir a API em detalhes, mas a seguir é uma breve descrição da funcionalidade e um bom ponto de partida para procurar:

\* Lib / dmllib.php: inserir, atualizar e recuperar dados do banco de dados. Estas são funções de nível mais micro, incluindo execute\_sql (), db\_uppercase (), modify\_database (), count\_records (), get\_record (), get\_field (), delete\_record (), insert\_record (), update\_record (), etc A partir de moodle 1.8.3 +, há cerca de 69 funções definidas neste arquivo de biblioteca (para mais informações consulte o site xref Moodle). O mais importante (ou mais frequentemente usados) funções nesta biblioteca são:

o get\_record () Obter um único registro como um “objeto”. Esta função é muito utilizado (ou seja, 1500 vezes referenciado em Moodle + código 1.8).

o get\_records () Obter um número de registros como um array de objetos. Esta função também é frequentemente usado (ie, referenciada 460 vezes + em Moodle 1.8. Código). Se a consulta for bem-sucedido e retorna pelo menos um registro, o valor de retorno é um array de objetos, um objeto para cada registro encontrado. A chave do array é o valor da primeira coluna do conjunto de resultados. O objeto associado com essa chave tem uma variável de membro para cada coluna dos resultados.

o get\_record\_sql () Obter um único registro como um objeto usando uma instrução SQL. A instrução SQL deve, normalmente, só retorna um registro. No modo de depuração você receberá um aviso se mais registros são retornados (a menos que você definir \$ expectmultiple para true). Em não-debug mode, ele só retorna o primeiro registro.

truque de mágica

Dica importante:

Ao acessar o banco de dados usam o banco de dados do Moodle API. O banco de dados API abstrai o banco de dados chamadas permitindo que os administradores do sistema para escolher qual banco de dados relacional funciona melhor para esta instalação. Não usar a API de banco de dados força o desenvolvedor criar consultas SQL para todos os bancos de dados relacionais Moodle suporta (MySQL, PostgreSQL, MSSQL e Oracle).

\* Lib / ddlib.php: criar, modificar ou excluir esquema do banco. No Moodle 1.8 que contém 27 funções. Esta biblioteca inclui funções como table\_column (), table\_exists (), field\_exists (), index\_exists (), find\_index\_name (), create\_table (), rename\_table (), add\_field (), drop\_field (), rename\_field (), add\_key (), drop\_key (), etc Algumas das funções mais importantes ou mais frequentemente utilizadas neste arquivo de biblioteca são:

o table\_column () Esta função irá adicionar um novo campo a uma tabela, ou alterar um já existente (se oldfield é definido). Foi referenciada 850 vezes + em códigos Moodle. Aviso: Tenha cuidado em chaves primárias em tabelas diferentes, como esta função vai comer auto\_increments.

o create\_table () Esta função irá criar a tabela passada como um argumento com todos os seus campos / chaves / índices / seqüências – tudo com base no objeto XMLDb. Como você poderia imaginar, há também DROP\_TABLE (), rename\_table (), add\_field (), drop\_field (), rename\_field (), e change\_field\_type () funções. Todos estão relacionados com a manipulação do esquema do banco de dados.

\* Lib / accesslib.php: Este é um arquivo de biblioteca novo criado a partir de Moodle 1.7. Cobre principalmente contexto, papéis e funções relacionadas com permissão. No Moodle 1.8.x há cerca de 80 funções definidas nesse arquivo accesslib.php. Funções mais utilizadas são:

o require\_capability () Verifica se há uma afirmação capacidade de ser verdadeiro. Se não for seguida, a página é terminada perfeitamente com uma mensagem de erro padrão.

o has\_capability () Você verá esta função que está sendo referenciado um lote (no Moodle 1.8.3 +, pelo menos, 915 vezes; ver o site do Moodle xref para mais detalhes). Esta função retorna onde o usuário atual tem a capacidade de executar uma função. Por exemplo, podemos fazer has\_capability (mod / forum: ? Replypost ‘, \$ contexto) no fórum. Esta é uma função recursiva, pelo caminho.

o get\_context\_instance () Obter a instância do contexto como um objeto. Esta função é frequentemente usado e irá criar a instância do contexto se não existe ainda.

o create\_role () e delete\_role () Cria um papel e exclui um papel (assim como limpá-lo depois)

\* Lib / blocklib.php: Em suma, este arquivo de biblioteca inclui todas as funções necessárias para usar blocos em uma página do curso típico. A partir do Moodle 1.8.3 +, há cerca de 27 funções incluídas neste arquivo. Alguns dos mais importantes (ou frequentemente utilizado) funções incluem:

o block\_instance () Carrega os arquivos de classe necessários para um bloco. Sempre que você quiser carregar um bloco, use esta função em primeiro lugar.

o blocks\_have\_content () Esta função irá aceitar uma matriz de instâncias de blocos e verifica se algum deles tem conteúdo para mostrar. Irá retornar verdadeiro ou falso. Ele está sendo referenciado no curso // formato / scorm / format.php, / admin / settings.php, e outros lugares.

o block\_print\_group () Imprime um grupo de blocos em uma página. Parâmetros passados por referência para a velocidade, não são modificados.

\* Lib / formlib.php: Este é um arquivo de biblioteca muito importante no Moodle 1.8. Esta é a biblioteca de classes para a criação de formas no Moodle, com base em QuickForms PEAR. Quer saber mais sobre as características de formlib? Adicionais úteis os recursos relacionados são de uso Formlib, definição de formulário Formlib e Validação Formlib.

Informações também estão disponíveis sobre as alterações mais recentes para os arquivos de biblioteca mencionados acima.

truque de mágica

Dica importante:

Saber para onde olhar eo que funciona para uso se tornará mais fácil com o tempo e familiaridade com o codebase Moodle. Pode parecer frustrante a princípio – mas seja paciente!

Selecionando arquivos de biblioteca para incluir

Ao incluir as bibliotecas do núcleo do Moodle, é melhor seguir a prática de menos é mais: incluindo apenas as bibliotecas que contêm as funções que você precisa para que o programa funcione. Sempre use a função require\_once () para evitar a inclusão de arquivos que podem já ter sido

incluída em outras partes do código do Moodle. O arquivo mais comuns para incluir é config.php. Este arquivo configura o global \$CFG variável (mais cerca de US \$ CFG para vir na Unidade 5 Parte C), bem como inclui outros arquivos essenciais. Outra prática comum é para armazenar todas as funções comuns de um bloco ou módulo em um arquivo chamado arquivo de biblioteca lib.php.

Este é apenas um breve resumo e os arquivos de biblioteca acima cobrem mais funcionalidade do que é sugerido pela lista acima. Existem muitas outras funções disponíveis através da API.

Exemplo trechos de código usando funções de Três arquivos de biblioteca principal

Exemplo 1

Objetivo: Implementar uma instrução if para ver se um usuário com 'joe' username existe no banco de dados do usuário.

Passo 1: Analisar a tarefa. Como a tarefa é lidar com o banco de dados do usuário, olhamos para o / lib / dml.php procurar funções úteis que podemos usar para recuperar dados do usuário.

Encontramos a função "record\_exists ()".

Passo 2: Implementar os trechos de código:

```
global $CFG;
require_once ($CFG->dirroot . 'config.php');
require_once ($CFG->dirroot . '/lib/dml.php');

if (record_exists ('user', 'username', 'joe')) {
    print 'joe é um nome de usuário';
}
else {
    print 'joe não é um nome de usuário';
}
```

Passo 3: Testar o código para ver se funciona. Métodos de ensaio são abordados com mais profundidade na Unidade 8.

topo

Exemplo 2

Objetivo: Imprimir um sim / não caixa de combinação para um formulário HTML com o nome do elemento "ativo", "não" como selecionados.

Passo 1: Analisar a tarefa. Como a tarefa é lidar com a saída HTML, olhamos para o / lib / weblib.php procurar funções úteis que nos ajudam a alcançar o nosso objectivo. Nós encontramos 'choose\_from\_menu ()' "a função (em torno de linha # 642).

Passo 2: Implementar os trechos de código:

```
global $CFG;
require_once ($CFG->dirroot . 'config.php');
require_once ($CFG->dirroot . '/lib/weblib.php');

$options = array ();
$options [0] = get_string ('não');
$options [1] = get_string ("sim");

choose_from_menu ($options, 'ativo', 'não');
```

Passo 3: Teste o código.

Exemplo 3

Objetivo: Limpeza e desinfecção de alguns dados apresentados a partir de um formulário e inserir os dados em tabela 'mdl\_notes' onde 'mdl\_' é o prefixo do banco de dados. Assumindo que "notas" é um textarea e 'url' é uma caixa de texto.



Passo 1: Analisar a tarefa. A tarefa é lidar com (1) limpeza de dados enviados de um formulário, e (b) inserir dados em uma tabela Moodle. Percebemos que, provavelmente, ter que olhar para o / lib weblib.php / (para dados apresentados de uma forma) e / lib moodlelib.php / (para limpeza de dados), e / lib dmllib.php / (para inserir dados no tabela Logo, temos que lidar com três arquivos de biblioteca: weblib.php função, moodlelib.php e dmllib.php Encontramos a data\_submitted (função) em weblib.php (em torno de linha # 285), e clean\_param (). em moodlelib.php (em torno de linha # 336) e, finalmente, o insert\_record () em dmllib.php. Se queremos lidar com o código escrito em uma “Moodle” mais uma forma, poderíamos usar get\_string () (retorna um sequência localizada), e print\_string () (imprimir uma string traduzida) funciona para mostrar o sucesso / erro. Desta forma, seria permitir que a mensagem de erro a ser exibido em diferentes idiomas (se a string é definida em um pacote de idioma particular).

Passo 2: Implementar os trechos de código:

```
global $CFG;
require_once ($CFG->dirroot . 'config.php');
require_once ($CFG->dirroot . '/lib/dmllib.php');
require_once ($CFG->dirroot . '/lib/weblib.php');
require_once ($CFG->dirroot . '/lib/moodlelib.php');

if ($dados = data_submitted ()) {
if (! empty ($data-> notas) & &! empty ($data-> url)) {
// Construir um novo objeto para inserir no banco de dados
$insert = new stdClass;
$insert-> notas = clean_param ($data-> notas, PARAM_CLEAN);
$insert-> url = clean_param ($data-> url, PARAM_URL);
if (insert_record ('notas', $insert)) {
print_string ('notesinserted');
}
else {
erro (get_string ('insertnoteserror'));
}
}
else {
erro (get_string ('missingrequireddata'));
}
}
else {
erro (get_string ('nodatasubmitted'));
}
```

Passo 3: Teste o código.

topo

“To Do” Lista

1. Abra seu editor de PHP favorito e percorrer a moodlelib.php agora.

truque de mágica

Dica importante:

A “busca” recurso no site xref é por vezes difícil de encontrar e utilizar. O link de pesquisa está localizada no canto superior direito do site e quando você “mouse-over” no link, você verá um embutido janela pop-up de opções (função de classe, variável, constante, e tabela).

2.

Ir para xref (<http://xref.moodle.org/>) e percorrer o mesmo arquivo (moodlelib.php). Desta vez, você

será capaz de clicar em funções ou variáveis imediatamente e saber onde eles foram definidos ou referenciados. Este site pode ser particularmente útil quando você tenta percorrer todas as principais funções cobertas de moodlelib.php.

captura de tela de xref

Este “recurso de busca” também irá gravar o seu histórico de pesquisa. Se você clicar em “Histórico de pesquisa +” link link, você vai ver todo o seu histórico de busca neste site:

screen shot xref

3.

Escolha um arquivo não abrangidos em discussão esta Unidade do diretório / lib e procurar o código para determinar o que ele faz.

4.

Navegue em weblib.php xref.moodle.org para ter uma idéia do que funções são abordados neste arquivo de biblioteca.

5.

Escreva um trecho simples que listar todos os cursos (com a sua id, nome abreviado, e fullname) que foram criados entre 2006/10/01 a 2007/12/01. (Dica: você pode precisar usar uma função na biblioteca moodlelib.php para converter uma data para um timestamp unix.) A saída deve estar usando o get\_string () função (em vez de usar apenas eco ou impressão para que esteja em conformidade com o Moodle padrão (para tornar mais fácil para outros idiomas). Envie seu trecho como um arquivo de texto na área de atribuição designada.

6.

Compartilhar seus comentários sobre as bibliotecas Moodle no fórum de discussão designado.

7.

Faça o teste. Duas tentativas são permitidas.

8. Adicionar à discussão sobre bibliotecas Moodle.Ouvir

Ler foneticamente

Ouvir

Ler foneticamente

## Unidade 6: Visão geral de práticas de programação

Visão geral | Unidade 6 “fazer” lista

Recursos externos abrem em novas janelas do navegador.

Há desafios definitiva para criação e manutenção de código criado por programadores ao redor do mundo que podem ser acostumados aos seus padrões de programação própria. Por isso não deveria ser surpreendente que o Moodle se desenvolveu diretrizes de codificação para agilizar a integração de código a partir de muitas fontes. Além da leitura atribuído sobre as orientações Moodle codificação, como você se prepara para escrever seus programas Moodle própria, lembre-se destas dicas rápidas:

- \* Use quatro espaços em vez da tecla Tab. Isso pode ser facilmente realizado por configurar corretamente o seu editor como você fez na Unidade 2 para o Eclipse.

- \* Os nomes das variáveis devem ser descritivos e úteis

- \* Escreva seu código para ser lido em vez de escrever comentários para explicar o código mal escrito.

Qualquer recurso boa segurança em primeiro lugar cobrir aspectos de engenharia social, porque muitas falhas de segurança são um resultado da falta de política ou protocolo seguinte política

existente. Este aparelho só brevemente cobrir aspectos de engenharia social – nem todos os aspectos possíveis de segurança. Há muitos recursos de qualidade por aí que vai fazer um trabalho muito melhor (de referência anexo) do que poderia fazer aqui. Por este motivo vamos nos concentrar em como gerenciar riscos envolvidos com o banco de dados orientado aplicações web. Também faremos uma breve cobrir outras questões de segurança e incluem algumas das melhores práticas de segurança.

“A grande mentira da segurança do computador é que a segurança melhora através da imposição de senhas complexas em usuários. Na vida real, as pessoas escrevem para baixo qualquer coisa que não me lembro. A segurança é aumentada através da concepção para a forma como os humanos realmente se comportam.” – Jakob Nielsen

Como a maioria dos problemas de segurança baseados em web ocorrer?

Erros na programação

Dando permissão demais para um usuário ou não restringir a permissão do usuário: Isso acontece quando não o fizemos, ou não pode, adequadamente verificar se um usuário tem a autoridade para executar um comando. Isto significa que os usuários não devem ser capaz de editar conta de outro usuário, a menos que tenha privilégios administrativos. Ou um usuário não deve ser capaz de mudar o curso, a menos que eles são os professores para esse curso ou que estão um administrador ou outro usuário que deve ter esse privilégio.

Erros de lógica geral: Há erros de lógica geral que parecem seguir os passos adequados, mas talvez um boolean condicional é invertida ou indevidamente negada. Isto é, que descemos do ramo errado da execução.

Erros no tratamento dos dados

Indevidamente manipulação de dados: Essencialmente não há dados provenientes de usuários devem sempre ser confiável. O que isto significa é que, independentemente de quem é nosso usuário atual é que devemos tratar os dados que entram como se pode ter conteúdo malicioso. Devemos dados sempre adequadamente limpa e filtro proveniente do usuário. Dados armazenados no banco de dados foi, provavelmente, inicialmente apresentado por um usuário para que os dados precisam ser tratadas corretamente quando reexibida para os usuários.

Falta de fundição / tipo de verificação: Esta é essencialmente uma extensão da manipulação de dados indevidamente. Quando estamos esperando um número como entrada do usuário devemos garantir que um número é fornecido. Sempre que possível, dados numéricos deve ser usado porque é extremamente fácil de limpar. No entanto, ainda precisamos fazer verificações nível apropriado de usuário para garantir que eles tenham o direito de ver ou modificar dados.

Expondo muita informação: – Alguns sistemas têm sido referidos como “lábios solta” quando eles expõem as informações demais. Um exemplo comum é um script de login que pode responder com “senha incorreta para ‘Joe’ usuário” este sistema é solto lábios porque alerta o usuário que ‘Joe’ é uma conta de usuário válida. A melhor mensagem seria “username ou senha incorreta” isso proporciona um usuário válido com informações sobre o que correu mal e uma pessoa tentando cortar um login com essencialmente nenhuma informação. Na maioria das vezes demasiada informação é exposta através de mensagens de erro, mas, infelizmente é frequentemente o caso que muito pouca informação é retornada. Seu script só deve retornar informações suficientes, e não mais, para o usuário entender o que aconteceu para que eles possam corrigir seu erro.

Ao manusear os dados de um usuário que deseja aplicar a filtragem mais apertado possível. Você quer deixar apenas os dados que você deseja passar. Nem mais nem menos. Se você está esperando um número, assegurar que os dados fornecidos é de fato um número. Se você está esperando uma sequência alfanumérica, garantir que é o que é fornecido.

A API sólido e robusto terá formas sólidas para lidar com dados de entrada. Infelizmente às vezes você pode não ser capaz de filtrar os dados de forma muito rigorosa. Este é o lugar onde ela se torna

extremamente importante que você trata os dados de forma adequada antes de tentar fazer qualquer coisa com o banco de dados envolvendo os dados, ou tratar adequadamente os dados antes de reexibição no navegador usuários.

Adequadamente a manipulação de dados envolve não apenas a manipulação de dados de entrada, mas também devidamente manipulação de dados de saída.

#### Erros no Confiança

Confiando em dados de um usuário não-confiável: Como cobertos acima, você não deve confiar em dados provenientes de usuários. Assim, a declaração pode ser de forma mais sucinta “dados confiança de um usuário.” Isto pode parecer demasiado severa ou restritiva, mas a razão para não confiar usuários é que uma pessoa indesejada pode obter acesso a uma conta de usuário. Ou seja, a pessoa do outro lado pode não ser quem você pensa que é.

Dados confiando do banco de dados: Isso já foi abordado também. Um erro comum é a suposição de que a entrada do usuário foi limpo antes de ser colocado no banco de dados assim que é seguro quando se trata de volta a partir do banco de dados. Esta é uma suposição extremamente perigoso como o usuário pode ter postado um código malicioso que não foi executado inicialmente devido a um manuseamento adequado, mas agora se assumirmos que é seguro o exploit original ainda pode ser possível. A razão para isto é que, quando inicialmente tratados os dados que podem ter protegido contra o código malicioso, mas o código malicioso ainda está presente nos dados. Ou seja, em vez de executar o código malicioso que temos armazenados no banco de dados. Agora, se mais tarde, assumir que é seguro, então podemos inadvertidamente executar o código malicioso.

Por que você não deve confiar usuários

Isto pode soar como uma dura política ou procedimento, mas isto é uma suposição justa e é uma boa prática.

- \* Os usuários cometem erros. Às vezes fazemos coisas que não temos a intenção de fazer

- \* Um usuário pode não ser quem você acha que eles são. Pode ser que o usuário do outro lado do computador podem não ser quem você acha que eles são. Isso pode acontecer por várias razões que estão fora do escopo deste documento.

- \* Um usuário pode precisar de educar os alunos sobre exploits. It é plausível que um usuário pode ter de educar seus alunos sobre as façanhas de segurança comum. Durante este processo, eles podem precisar para postar código que se não for devidamente tratado seria malicioso ou ter consequências inesperadas. O instrutor pode postar este código, para que você não quer confiar em que a informação fornecida pelo instrutor é seguro.

Confiando usuários leva a vulnerabilidades: Ao confiar seus usuários você está colocando a segurança de todo o seu site em sua base de usuários inteira. Se isso não deixá-lo nervoso, talvez você deve considerar uma outra profissão, e certamente não o sistema de administração! Quando você confia em usuários que você está assumindo que eles não vão fazer qualquer coisa perigosa ou malicioso intencional ou acidentalmente. Por exemplo, uma falha que permite que permissões elevadas é uma vulnerabilidade que poderia permitir que um estudante para obter privilégios professor e um professor para adquirir privilégios de administrador. Uma vez que o sistema é penetrada, os problemas parecem aumentar e falhas de segurança vai se tornar cada vez mais grave.

topo

## Cross-Site Scripting (XSS)

O que é XSS?

Exploits Cross-Site Scripting são tipicamente o resultado de confiar em dados de uma fonte e apresentá-lo para outro usuário sem a devida digitalização e manipulação de dados iniciais ou devidamente filtragem dos dados antes de enviá-lo para um segundo usuário a ser exibido.

Por que devemos proteger contra XSS, ou o que são algumas implicações de um exploit de XSS?

Muitos vendedores minimizar a gravidade dos exploits XSS. Isto é bastante lamentável, porque, como veremos XSS representar problemas de segurança para outros sites, bem como informações pessoais de um usuário é armazenado em seu computador. Abaixo estão alguns resultados potenciais de empregar XSS:

- \* Roubo Cookie: Obtenção de dados de cookies de sites em um domínio diferente. Este pode expor informações pessoais armazenadas em um cookie emitido por um site em um domínio diferente. Esta é uma evasão direta do DOM e pode expor informações pessoais do usuário ou dar acesso a informações armazenadas em um cookie de outro site.

- Conta \* seqüestro: “Logging in” para dar conta de um usuário diferente. Isso poderia ser conseguido por roubar as informações do cookie de outro site que emprega as conexões persistentes. Ou talvez um usuário altera uma conta para um aplicativo web especial para torná-lo tão fácil de login.

- \* Alteração de configurações de aplicativos web: Alterando a senha do usuário ou preferências de um determinado site tornando-o indisponível ou inutilizado pelo usuário correto.

- \* Exploit de um problema de segurança web browser para acessar um sistema do usuário operacional subjacente: Esta é uma questão de explorar dupla aqui. Empregando um problema de segurança conhecido por um navegador da Web em particular é possível que um usuário utilizando um ataque XSS pode ganhar privilégios elevados para o sistema operacional subjacente de que o browser está sendo executado.

Como podemos proteger contra XSS?

Há dois pontos-chave importantes na proteção contra XSS, e outra que alguns podem argumentar não está relacionado com XSS per se, mas não aliviar alguns dos problemas potenciais de segurança de alguém explorando uma vulnerabilidade XSS. Os dois pontos principais são:

- \* A validação da entrada: Ao validar entrada podemos tentar reconhecer dados que podem ter consequências inesperadas quando recuperados do banco de dados e enviados para o navegador ou enviadas diretamente para um navegador. Há várias opções a tomar quando se tenta manipular a entrada do usuário. Podemos nos recusar a fazer qualquer coisa para os dados que pareça suspeito, ou podemos escolher para filtrar os dados recebidos. Desde exploits XSS não são um alvo estático, é fundamental que haja uma única função para lidar com a identificação e limpeza de dados potencialmente problemáticos. (Moodle criou o **clean\_param function ()** para este fim.) Isso faz com que o código gerenciável como novas explorações são inventados há um lugar para update para garantir que todo o código-base é seguro. Isto, obviamente, pressupõe que todos estão devidamente codificação e utilizar as funções API apropriada para qualquer aplicação que está programando para.

- \* Codificação de dados antes de enviar para um usuário: através da codificação de dados antes de ser enviado para o usuário, podemos garantir que determinado texto que pode ser interpretado como JavaScript válida (ou código de script) está devidamente escapou para que ele seja mostrado como texto e não interpretado como marcação válido para ser executado. Isso evita que o usuário de execução accidental de algum tipo de script. Veja as regras gerais do Moodle Coding Guidelines.

O outro método é a verificação de um token privado, e realmente protege contra a “falsificação Cross Site Request” (CSRF), mas este é muitas vezes confundido com XSS. (Moodle usa **sesskey ()** como seu símbolo privada.) A falsificação Cross Site Request é quando um usuário artesanato uma URL que aparece para fazer uma coisa, mas quando outro usuário clica nele algum comando ou procedimento será executado. Por exemplo Joe poderia criar um post polêmico em um fórum ou algo que provavelmente causaria usuários a clicar em um URL crafted que tira proveito de uma vulnerabilidade CSRF em um aplicativo. Agora vamos supor que Maria é ler o post de Joe e clica na URL. Se um aplicativo é vulnerável a ataques CSRF, Maria pode ter acabado de excluiu um arquivo ou inconscientemente alterou sua preferências do usuário. Para se proteger contra ataques CSRF a aplicação pode gerar tokens privada que deve ser apresentado juntamente com pedidos de

atualização, criar ou excluir dados. O token privada é gerado para cada sessão e é verificada a qualquer momento que o usuário tenta alterar dados. Veja a seção Problemas de Segurança do Moodle Coding Guidelines.

Como podemos proteger contra XSS no Moodle?

\* A validação de entrada: lib comentário / moodlelib.php para obter informações apropriadas

PARAM\_TYPE

o required\_param (\$ parâmetro, PARAM\_TYPE);

Use esta função quando um parâmetro deve estar presente. \$ Parâmetro é esperado para ser um nome de parâmetro que será obtido via POST ou GET em que Moodle ordem. A API irá lançar um erro e parar a execução do script se o parâmetro não está presente. Use uma PARAM\_TYPE apropriado para filtrar a variável.

o optional\_param (\$ parâmetro, \$ default, PARAM\_TYPE);

Use esta função quando um parâmetro pode estar presente. \$ Parâmetro é esperado para ser um nome de parâmetro que será obtido via POST ou GET, nessa ordem. Se a variável não está presente, então \$ padrão é retornado. Use uma PARAM\_TYPE apropriado para filtrar a variável.

o clean\_param (\$ variável, PARAM\_TYPE);

Use esta função para limpar uma variável que já está definida no código.

\*

Codificação de dados: Ver documentação do desenvolvedor Moodle Output Functions.

\*

Símbolo privado:

o sesskey ()

+ Certifique-se que o parâmetro sesskey () é definido sob qualquer forma que irá resultar em que o usuário de fazer alterações de dados no banco de dados. Estes formulários devem usar o método POST e não GET sempre que possível.

+ Echo '<input type="hidden" name="sesskey" value="'.sesskey().'" />';

+ Echo '<input type="hidden" name="sesskey" value="'.USER-> sesskey. '" />';

o confirm\_sesskey ()

+ Ao processar os dados do formulário e seu script irá alterar quaisquer dados no banco de dados sempre garantir para confirmar que o parâmetro chave de sessão está presente e a chave de sessão corresponde a chave de sessão atual. Use confirm\_sesskey () que irá lançar um erro e parar a execução se os valores não coincidem.

Injeções de SQL

Quais são as injeções de SQL?

Injeções de SQL são uma forma de explorar a segurança do banco de dados onde os comandos arbitrários pode ser executado. Isso geralmente ocorre quando a entrada do usuário não é devidamente filtrada ou quando os dados não são fortemente tipados.

Um exemplo de injeção de SQL seria site de uma empresa de cartão de crédito, usando seu nome de usuário e senha para consultar o seu número de cartão de crédito, número de segurança social, e endereço para correspondência. Se o programador confia que o usuário digite as informações sem limpar o nome de usuário ou senha e decide incluir o nome de usuário e senha diretamente para o código php que cria a consulta SQL, o código de consulta SQL é semelhante a:

\$ Sql = "SELECT creditcardnum, ssn, endereço

FROM userdata

Onde username = '". \$ Username'".

E password = '". \$ Password . '";

Agora vamos supor que um hacker mal intencionado enviou as seguintes informações para o programa:

Username = teste  
Password = blah 'ou '1' = '1

Estes parâmetros criar uma consulta SQL:

```
SELECIONE creditcardnum, ssn, endereço  
FROM userdata  
Onde username = 'blah'  
E password = 'blah' ou '1' '=' '1'
```

Declaração “ou” faz com que este para selecionar todos os registros na tabela de usuário – não apenas aquele com o nome de usuário e senha corretamente. Parece que hacker mal intencionado tem um monte de clientes de cartão de crédito agora!

Por que é importante para proteger contra injeções de SQL?

É importante para proteger contra injeções de SQL porque um usuário pode ser capaz de executar qualquer comando de banco de dados. Cenários possíveis incluem eliminação catastrófica do banco de dados inteiro, exposição de informações confidenciais do usuário, ou sutil difícil de detectar alterações como a adição de alguns pontos a uma atribuição da classificação.  
Como podemos proteger contra injeções de SQL?

Para se proteger contra injeções SQL devemos adequadamente filtro de texto do usuário e ser cuidado como usamos a entrada do usuário para gerar consultas SQL. Além disso, podemos fortemente tipo de consultas para que os parâmetros sejam adequadamente tratadas. Para filtrar texto que precisamos lidar corretamente com caracteres de escape para que eles não sejam mal interpretadas, permitindo que um usuário execute modificado ou várias consultas SQL. Dados fortemente digitando passado pelo usuário garante que os dados é do tipo que esperamos que ele seja. Ou seja, se estamos esperando um número do usuário então podemos fortemente tipo que variável para garantir que detém um número ao invés de uma string.  
Como podemos proteger contra SQL injeções no Moodle?

\* Entrada do usuário Filter – filtro de entrada do usuário adequadamente usando:  
o required\_param ()  
o optional\_param ()  
o clean\_param ()  
\*

Entrada do usuário fortemente tipo – Sempre que possível, o PARAM\_TYPE para o acima de três funções será este método aproximados.

topo

Dados Moodle Lavagem Parâmetros

O seguinte é uma visão geral do parâmetro de limpeza tipos para ser usado com required\_param (), optional\_param (), e clean\_param ().

- \* PARAM\_RAW: especifica um parâmetro que não é limpo ou processado de qualquer forma.
- \* PARAM\_CLEAN: obsoletos, por favor, tente usar um tipo mais específico de parâmetro.
- \* PARAM\_INT: apenas números inteiros, use quando esperando apenas números
- \* PARAM\_INTEGER: Alias para PARAM\_INT
- \* PARAM\_ALPHA: Contém apenas letras Inglês.
- \* PARAM\_ACTION: Alias para PARAM\_ALPHA, uso para diversas ações em formulários e URLs.
- \* PARAM\_FORMAT: Alias para PARAM\_ALPHA, use os nomes dos plugins, formatos, etc
- \* PARAM\_NOTAGS: Todas as tags HTML são retirados do texto. Não abuse desse tipo.
- \* PARAM\_MULTILANG: Sinônimo de PARAM\_TEXT.
- \* PARAM\_TEXT: texto simples Geral compatível com MultiLang filtro, não outras tags html.
- \* PARAM\_FILE: nome do arquivo de segurança, todos os chars perigosos são removidos, protege

contra o XSS, injeções de SQL e percursos de diretório.

\* PARAM\_PATH: nome do caminho seguro relativos, todos os chars perigosos são removidos, protege contra o XSS, injeções de SQL e percursos de diretório

nota o: A barra de liderança não é removido, letra de unidade janelas não é permitido

\* PARAM\_HOST: nome de domínio esperado totalmente qualificado (FQDN) ou um quad IPv4 pontilhado (endereço IP)

\* PARAM\_URL: esperado URL formatado corretamente.

\* PARAM\_LOCALURL: esperado URL formatado corretamente, assim como aquele que refere-se ao servidor local em si. NÃO ortogonal aos outros! Implica PARAM\_URL!

\* PARAM\_CLEANFILE: nome do arquivo seguro, todos os chars perigoso e regionais são removidos,

\* PARAM\_ALPHANUM: números esperados e apenas letras.

\* PARAM\_BOOL: converte a entrada em 0 ou 1, use para interruptores em formulários e urls.

\* PARAM\_CLEANHTML: limpa submetidos código HTML e remove barras

nota o: não se esqueça de addslashes () antes de armazenar no banco de dados!

\* PARAM\_ALPHAEXT: o mesmo conteúdo PARAM\_ALPHA mais os chars entre aspas: “/ - ” permitido, adequado para include () e require ()

\* PARAM\_SAFEDIR: nome do diretório seguro, adequado para include () e require ()

\* PARAM\_SEQUENCE: espera uma sequência de números como 8-1, 5, 6, 4, 6, 8, 9. Números e vírgulas apenas.

topo

Aspectos de segurança de funções do Moodle e Sistema de Capacidades

Começando com o Moodle versão 1.7 de um sistema de funções e capacidades foi introduzido para substituir o actual sistema de papéis fixos (por exemplo, os administradores, os criadores do curso, os professores edição, estudantes, etc.) Dos erros de segurança descritos anteriormente, alguns erros de lógica e alguns erros na confiança são mitigados por esse sistema de papéis, recursos e permissões.

A capacidade é normalmente alguma ação específica que Moodle pode ser executado por um usuário. Por exemplo, a capacidade de ‘moodle / curso: update’ se refere à ação de configurações de curso de atualização. Outra capacidade, “moodle / curso: viewhiddencourses ‘, se refere à ação de ver os cursos que foram escondidos.

Um papel é um conjunto nomeado de todos os recursos com permissões associados (por exemplo, estudante, moderador do fórum, professor assistente de professor).

A permissão descreve a capacidade de um papel a desempenhar uma determinada capacidade.

Permissões para um recurso específico de um papel específico são definidos dentro de um contexto ou “espaço” no Moodle. Por exemplo, um curso seria considerado um contexto. Os quatro permissões disponíveis para ser definido para uma capacidade de um papel em um contexto específico são:

1. CAP\_INHERIT
2. CAP\_ALLOW
3. CAP\_PREVENT
4. CAP\_PROHIBIT

Para entender como essas quatro permissões de trabalho, é necessário entender que os contextos são hierárquicas. De mais amplo ao mais específico, os contextos são:

1. CONTEXT\_SYSTEM
2. CONTEXT\_PERSONAL
3. CONTEXT\_USER
4. CONTEXT\_COURSECAT
5. CONTEXT\_COURSE



## 6. CONTEXT\_MODULE

## 7. CONTEXT\_BLOCK

Permissões não definida em um contexto para a capacidade de uma função são herdadas (CAP\_INHERIT) a partir de um contexto mais geral.

Por exemplo, se o papel aluno tem uma permissão CAP\_ALLOW para o 'moodle / curso: view' "capacidade em um contexto CONTEXT\_COURSE e os contextos subjacentes (CONTEXT\_MODULE CONTEXT\_BLOCK e) não ter sido" substituído " , em seguida, o papel também terá CAP\_ALLOW para 'moodle / curso: view' nesses contextos mais específicos.

A permissão CAP\_PREVENT negará capacidade de um usuário que nesse contexto e contextos mais específicos, a menos que substituído em tais contextos. É importante perceber que as permissões CAP\_PREVENT pode ser substituído em contextos mais específicos. Por exemplo, um papel de estudante pode ter CAP\_PREVENT em 'mod / forum: taxa' "a capacidade em um contexto claro, mas de um fórum específico (contexto módulo) têm CAP\_ALLOW. A permissão, no contexto específico "ganha".

A permissão CAP\_PROHIBIT também nega que a capacidade de um usuário no contexto em que está definido, mas a permissão em contextos mais específicos não podem ser substituídos.

topo

Permissões legado

Permissões legado no sistema Moodle papéis têm sido implementadas para manter a compatibilidade. Em versões do Moodle antes 1,7 havia cinco papéis estáticos:

1. Administradores
2. Criadores de curso
3. Professores
4. Estudantes
5. Convidados

No Moodle 1.7 e além dos mesmos papéis existem em uma instalação padrão do Moodle. Um administrador Moodle agora tem a capacidade de adicionar, excluir ou alterar qualquer permissão em sua instalação Moodle. Ao criar uma nova capacidade desenvolvedores definir o valor padrão para cada uma dessas funções padrão na seção legado, a fim de configurar as permissões padrão para uma instalação Moodle. Todas as funções não-padrão terá de ser configurado manualmente pelo administrador do Moodle quando um bloco ou um módulo com uma nova capacidade instalada.

Funções Moodle para as Funções e Capacidades

Existem várias funções importantes no Moodle nos papéis e capacidades do sistema.

A primeira linha de defesa em um script é o require\_login () para restringir o acesso a utilizadores registados. Esta função verifica primeiro que o usuário está conectado e, em seguida, opcionalmente, verifica se eles estão autorizados a estar em um determinado curso e / ou visualizar um módulo do curso particular.

Para verificar a permissão do usuário para um recurso em um determinado contexto requer um par de funções. O primeiro, get\_context\_instance (), retorna uma instância de contexto, como um objeto. A instância de contexto é um nível de contexto (por exemplo CONTEXT\_COURSE) e um ID de instância (por exemplo, um ID é claro).

Depois de ter a instância contexto que você pode exigir uma capacidade em um script usando o require\_capability function () ou você pode verificar se um usuário tem uma capacidade especial, antes de executar alguma ação com o has\_capability function (). Uma aplicação dessas funções pode ter esta aparência:

```
$ Context = get_context_instance (CONTEXT_COURSE, $ CourseID);
```

```

$ Require_capability ('moodle / curso: view', $ contexto);
// Algumas coisas que qualquer pessoa com moodle / curso:? Vista pode
// Fazer ou ver aqui
// Agora verificar se há uma capacidade mais restritiva
if (has_capability ('moodle / curso: update' contexto $) {
// Fazer alguma atualização para o curso
Else {}
error ('Você não tem as permissões corretas');
}

```

topo

## Programação para Acessibilidade

Como mencionado na introdução deste curso, os desenvolvedores do Moodle em todo o mundo têm feito progresso em direção à acessibilidade universal – em grande parte por causa do compromisso de fazer melhorias no código que melhorar a usabilidade. É imperativo que não os desenvolvedores de código Moodle futuro dismantelar os avanços significativos alcançados até à data e continuar a observar algumas práticas de programação usuários-chave que têm impacto.

Algumas dicas rápidas para manter em mente ao projetar elementos Moodle:

- \* Tanto quanto possível, evitar eventos que causam janelas pop-up.
- \* Use nomes ligação efectiva e etiquetas elemento de formulário
- Funções \* devem ser executável a partir de um teclado
- \* Nunca use cores para transmitir significado
- \* Evite elementos piscando ou animados
- \* Fornecer descritivo e relevante texto alternativo para gráficos ou imagens, lembrando que ele será lido em voz alta para os usuários com deficiência visual

## Em Desenvolvimento-Documentação do Código

Moodle usa PHPDocumentor para gerar documentação on-line sobre classes e funções comuns. É recomendável que você usar o formato PHPDocumentor ao documentar suas funções e classes. Isto permite que outros desenvolvedores para exibir informações sobre as funções que você criar on-line sem a necessidade de procurar o seu código. PHPDocumentor estilo também faz para limpar, fácil de ler código que um desenvolvedor pode usar para entender rapidamente o que uma função é suposto fazer eo que cada parâmetro é sem ler toda a função. Consulte a documentação do Moodle PHP.

topo

## “To Do” Lista

1.

Ler atentamente as Diretrizes para Codificação em Moodle.org-dando atenção especial para a seção Questões de Segurança.

2. Rever o Checklist Seção WebAIM 508 para sugestões importantes para alcançar a acessibilidade universal.

3.

Leia sobre os recursos Moodle para melhores práticas de segurança e addslashes (), que por vezes pode ser confuso.

4.

## Completar o Tutorial PHPDocumentor

5. Detectar a segurança, acessibilidade, documentação, orientação e erros de codificação do exemplo a seguir. Apresentar suas correções na caixa drop designado na seção desta Unidade na página prato principal. Copie o seu código alterado em um arquivo somente texto e enviar o

arquivo.

```
<? Php
/*
* Criado em 11 de dezembro de 2007
*
* Este script testa o usuário tem a ver claro
* Capacidade e, em seguida, verifica se a senha digitada
* Está correto. Se a senha estiver correta, ela imprime uma página web.
*/
require_once ('http://www.somesite.com/abadfile.php&#8217;);
required_param ($ id, PARAM_INTEGER);
required_param ($ senha, PARAM_TEXT);
required_param ($ CourseID, PARAM_INTEGER);
optional_param ($ sitename "Humboldt", PARAM_TEXT);
$ File = get_file_contents ($ _GET ['url']);
$ Context = get_context_instance (CONTEXT_COURSE, $ CourseID);
$ Require_capability ('moodle / curso: view', $ contexto);
if ($ senha! = 'thisistherightpassword') {
erro ("<blink> sua senha não está correta </ blink>");
Else {}
echo $ arquivo;
// Esta função faz algum além
block_addition função ($ a, $ b) {
return ($ a $ b +);
}
?>
```

6.

Faça o teste. Duas tentativas são permitidas.

7.

Adicione seus pensamentos para a discussão sobre acessibilidade.

## Unidade 7 – Parte A – Bloco Básico

### Uma parte – Bloco Básico

1. Criar um simples bloco seguinte tutorial do Moodle. O bloco será nomeado “simplehtml”.
2. Criar um arquivo de linguagem e de diretório.
3. Criar uma tabela de banco de dados em que nosso bloco irá inserir informações.

### Parte B – Formulários

1. Adicionar um formulário usando o Moodle biblioteca de formulários para inserir as informações na tabela
  - \* Definir uma nova forma.
  - \* Apresentar o novo formulário.
  - \* Adicione um cabeçalho.
  - \* Adicione o controle de acesso.
  - \* Adicione registro.
  - \* Adicione elementos de formulário.
  - \* Adicione as variáveis de estado.

- \* Finalize o formulário.
2. Processar o formulário.

#### Parte C – Refinar a Interface de Usuário

1. Adicionar links para cada página do bloco.
  2. Adicionar métodos para visualizar páginas.
  3. Refactor método de exibição.
- \* Adicionar título da página.
  - \* Mostrar a data
4. Exibição do texto.
  5. Mostrar link para o arquivo.
  6. Exibir a imagem.

#### Parte D – Adicionar recursos de edição

Adicionar recursos de edição que permitem aos usuários editar dados, incluindo

1. Excluir uma página
2. Adicionar um link para o arquivo delete.php
3. Criar o arquivo delete.php
4. Limpar todos os dados associados com o bloco com o `instance_delete()` função

#### Parte E – definir um papel para o Bloco

1. Adicionar verificação para o bloco para a capacidade
2. Adicionar um papel para a forma simplehtml
3. Adicionar uma função para editar
4. Adicionar uma função para apagar

Noções básicas de bloqueio | Formulários | Refine the User Interface | Adicionar recursos de edição | definir um papel

Hiperlinks será aberto em novas janelas do navegador.

Agora que nós cobrimos o básico sobre as bibliotecas Moodle, programação segura, seu ambiente de desenvolvimento, e como instalar e configurar o Moodle, vamos chegar à parte divertida – a codificação! Como você aprendeu na Unidade 5, o Moodle é modularizado, tanto quanto possível para permitir a programas de terceiros para integrar com o Moodle. A maneira mais simples de integrar um programa com o Moodle ou desenvolver novas funcionalidades é a de criar um bloco independente.

Na Unidade 7, você irá criar um bloco chamado simplehtml. Você vai começar com um tutorial Moodle.org que vai te ensinar como criar um bloco Moodle básico com funções mais utilizadas.

Ao completar o bloco simplehtml, você:

- \* Adicionar uma tabela do banco de dados para o Moodle
- \* Criar um formulário Moodle
- \* Adicione vários elementos HTML QuickForm a uma forma Moodle
- \* Processo de formar uma Moodle com vários elementos HTML QuickForm
- \* Inserir, atualizar e excluir dados de uma tabela do banco de dados no Moodle usando o formulário.

Alguns dos exemplos neste curso pode não parecer lógico ou útil no contexto eles são usados. Em geral, isso é intencional para ilustrar métodos diferentes. Devido à quantidade de detalhes neste tópico, é dividido em partes que devem ser completados em sequência, começando com esta Parte A, “Fundamentos Block.”

Criando um simples bloco

Siga o tutorial na documentação do site Moodle para criar um bloco simples. Testá-lo para

certificar-se que instala e funciona. Se precisar de ajuda, revisão “Unidade 4 -! Configurando Moodle para o Desenvolvimento” que abrange a instalação de blocos e módulos. A fim de corrigir alguns avisos PHP que irá aparecer a partir do código do Moodle tutorial você vai querer mudar a sua função de especialização para o seguinte.

```
especialização function () {  
if (! empty ($ this-> config-> title)) {  
$ This-> title = $ this-> config-> title;  
}  
if (empty ($ this-> config-> text)) {  
$ This-> config-> text =”;  
}  
}
```

Observe que esse código verifica se as variáveis de configuração foram definidos, se qualquer uma das variáveis não foi definido se atribui uma string vazia para essa variável ou não usá-lo. Isto reduz os avisos e segue boa prática de codificação de verificar se uma variável é instanciada antes de ser usado.

Verifique se o bloco que você criou usando o tutorial funciona instalando-o (consulte “Unidade 4 -! Configurando Moodle para o Desenvolvimento”) e adicionando o bloco para um curso de teste.

Introdução à Linguagem do Moodle e Internacionalização

directoryNow linguagem arquivo que o simples bloco foi criado, é hora de expandi-lo para fazer mais coisas. Como discutido na Unidade 5, um dos diretórios mais importantes para a customização do Moodle é o diretório lang. Cada bloco pode ter seu diretório própria língua. No diretório bloco criar um diretório lang (clique direito sobre o diretório simplehtml, em seguida, selecione Novo> Pasta).

Em seguida, crie um novo diretório dentro do diretório lang chamado en\_utf8 (Inglês codificados usando a codificação de caracteres utf8). Agora adicione um novo arquivo PHP com o nome block\_simplehtml.php para o diretório simplehtml/lang/en\_utf8.

Este arquivo conterá todas as cordas estáticas que o bloco simplehtml vai usar. Mais tarde, se você quiser traduzir a sequência para outro idioma que você pode adicionar outro diretório dentro lang para o idioma que estão traduzindo. Diretórios exemplo utf8 pode ser encontrado no diretório de sites lang se você tiver instalado outros pacotes de idioma. Mais sobre traduções.

Agora que o arquivo de idioma é criado, uma nova sequência pode ser adicionado ao Moodle para o bloco usando algo como o código de exemplo abaixo – que substitui o “nome da string ‘com uma chave válida array PHP eo’ Texto para a cadeia ‘ com uma sequência de PHP:

```
$ String’name da string ‘=’ Texto para a string ‘;
```

Prática, adicionando a linha abaixo ao seu arquivo block\_simplehtml.php.

```
$ String’blockname Bloco ‘=’ HTML Simples;
```

Para recuperar essa string dentro do bloco você usaria a função get\_string Moodle () encontradas em lib / moodlelib.php. Para obter a sequência de título em um uso variável:

```
$ This-> title = get_string (‘blockname’, ‘block_simplehtml’);
```

Substituir a linha na definição de classe para block\_simplehtml.php que atribui um valor de \$ this-> título com a linha de código acima.

Criar uma tabela de banco de dados para o Bloco

Com o arquivo de idioma começou é hora de passar a usar o Editor XMLDb para desenhar e criar a tabela banco de dados que irá armazenar os dados do formulário que você irá criar mais tarde. [Se você quiser mais detalhes sobre o Editor XMLDb, consulte o Apêndice B.] O usuário de entrada de

oito peças de informação para a forma simplehtml:

1. Texto a ser exibido com o link para a visualização das informações de forma
2. Displayable HTML ou texto
3. Formato HTML área
4. Um nome de arquivo
5. Um “sim ou não” questão
6. A escolha de imagem
7. HTML ou texto simples a partir de uma área de texto
8. Um carimbo de data e hora

ponto de exclamação

Dica importante!

Evitar o uso de enum porque nem todos os bancos de dados de suporte deste tipo. É melhor usar um inteiro que é codificada para um valor string. Inteiros também escala melhor do que enums.

Para começar, crie a estrutura de diretório correto em Moodle. Adicionar um diretório arquivo chamado db sob seu diretório bloco principal. Próximo login para Moodle e navegue até o editor XMLDb, localizado no “Site da administração” block> Diversos> XMLDb Editor. O editor XMLDb é uma interface gráfica para ajudar a gerar os arquivos xml que o Moodle utiliza para criar e manter banco de dados tabelas. Quando você chegar ao editor XMLDb você vai ver todos os blocos, módulos e outras áreas que têm diretórios db. Aqueles que mostram um link já tem tabelas definidas que podem ser carregados e modificados. Encontrar os blocos / simplehtml / db e clique em Criar.

Atualizar o navegador Eclipse. Agora, o diretório db tem um arquivo install.xml com algumas informações básicas sobre o seu bloco e uma tabela chamada simplehtml. No Moodle você vai ver agora o bloco / simplehtml / db destacado em verde e os de carga e Delete será ligada. Clique em Load. Agora você vai ver o Editar e Unload texto vinculado. Clique em Editar.

A tabela simplehtml deve ser modificado para incluir as informações inseridas no formulário: clique no link Editar ao lado da mesa simplehtml para mostrar a definição que é criado por padrão para a tabela. Campo id vai estar lá e uma chave primária que liga para o campo id. Renomear a tabela para block\_simplehtml. Alterar a comentários para esta tabela aqui também. Quaisquer mudanças no campo de comentários requer um clique no botão Alterar para que as alterações sejam gravadas.

Agora adicione os seguintes campos:

1. blockid
2. pagetitle
3. displaytext
4. formato
5. nome do arquivo
6. quadro
7. descrição
8. displaypicture
9. DisplayDate

topo

blockid

editar campo para blockidThe blockid é uma chave estrangeira que referencia a tabela do bloco. Isso vai ser usado para unir linhas nossos dados para a tabela de bloqueio. A tabela bloco casas informações bloco genérico para instâncias de bloco instalado. Para adicionar o campo blockid:

1. Clique no botão “New Field” link
2. Altere o nome para blockid
3. Insira um comentário sobre o campo

4. Definir o tipo de int
5. Digite um comprimento de 10
6. Set sem sinal para unsigned
7. SET NOT NULL para NOT NULL
8. Set Sequence para Não
9. Digite 0 para o padrão
10. Clique no botão Alterar.

#### pagetitle

editar campo para pagetitle O pagetitle é simplesmente texto que será usado para ligações para esta página simplehtml. Usando Temas Moodle padrão, blocos Moodle não consegue lidar com mais de 25 caracteres sem embrulho. Para adicionar o campo pagetitle:

1. Clique no botão “New Field” link
2. Altere o nome para pagetitle
3. Insira um comentário sobre o campo
4. Definir o tipo para char
5. Digite um comprimento de 25
6. SET NOT NULL para NOT NULL
7. Clique no botão Alterar.

#### displaytext

editar campo para displaytext O campo displaytext será exibido como texto na página simplehtml, potencialmente como HTML. Esta será uma área de texto HTML onde o usuário deve ter permissão para escrever como muito texto ou HTML como desejada – o que significa que deve ser definido como um “grande” campo de texto de comprimento. Para adicionar o campo displaytext:

1. Clique no botão “New Field” link
2. Altere o nome para displaytext
3. Insira um comentário sobre o campo
4. Definir o tipo de texto
5. Digite um comprimento de grandes
6. SET NOT NULL para NOT NULL
7. Clique no botão Alterar.

#### formato

editar campo para formatar o campo formato irá conter o valor inteiro que representa a formatar o texto foi inserido dentro Para adicionar o campo formato:

1. Clique no botão “New Field” link
2. Mudar o nome para o formato
3. Insira um comentário sobre o campo
4. Definir o tipo de int
5. Digite um comprimento de 3
6. Set sem sinal para unsigned
7. SET NOT NULL para NOT NULL
8. Digite 0 para o padrão
9. Clique no botão Alterar.

#### nome do arquivo

editar campo para filename O campo de nome de arquivo armazena o nome de um arquivo de posts do instrutor na página simplehtml, que os alunos podem então baixar ou ver. Para adicionar o campo nome do arquivo:

1. Clique no botão “New Field” link
2. Altere o nome para nome de arquivo
3. Insira um comentário sobre o campo

4. Definir o tipo para char
5. Digite um comprimento de 255
6. SET NOT NULL para NOT NULL
7. Clique no botão Alterar.

quadro

editar campo para foto A URL para a imagem escolhida do usuário está em uma variável estática predefinidos que associa um número a uma imagem. Isso permitirá que a adição de imagens mais tarde, se necessário, ou para mudar as imagens URLs atual 'sem atualizar o banco de dados. Porque só o número é armazenado no campo de banco de dados deve ser criado como um int em vez de um char. Para adicionar o campo de imagem:

1. Clique no botão "New Field" link
2. Mudar o nome para a imagem
3. Insira um comentário sobre o campo
4. Definir o tipo de int
5. Digite um comprimento de 2
6. Set sem sinal para unsigned
7. SET NOT NULL para NOT NULL
8. Set Sequence para Não
9. Digite 0 para o padrão
10. Clique no botão Alterar.

topo

descrição

editar campo para descrição Adicione o campo de descrição que será exibida na página. Esta será uma área de texto HTML onde o usuário tem permissão para escrever tanto quanto desejado por isso é um campo de texto "grandes". Para adicionar o campo de descrição:

1. Clique no botão "New Field" link
2. Altere o nome para descrição
3. Insira um comentário sobre o campo
4. Definir o tipo de texto
5. Digite um comprimento de grandes
6. SET NOT NULL para NOT NULL
7. Clique no botão Alterar.

displaypicture

editar campo para displaypicture O campo displaypicture será o "sim" ou "não" a questão no formulário. Este é o melhor representado como um 'int'. Para adicionar o campo displaypicture:

1. Clique no botão "New Field" link
2. Altere o nome para displaypicture
3. Insira um comentário sobre o campo
4. Definir o tipo de int
5. Digite um comprimento de 1
6. Set sem sinal para unsigned
7. SET NOT NULL para NOT NULL
8. Set Sequence para Não
9. Digite 0 para o padrão
10. Clique no botão Alterar.

DisplayDate

editar campo para DisplayDate O campo final para adicionar à mesa é o DisplayDate. É um timestamp UNIX que é um valor inteiro. Para adicionar o campo DisplayDate:

1. Clique no botão "New Field" link



2. Altere o nome para DisplayDate
3. Insira um comentário sobre o campo
4. Definir o tipo de int
5. Digite um comprimento de 10
6. Set sem sinal para unsigned
7. SET NOT NULL para NOT NULL
8. Set Sequence para Não
9. Digite 0 para o padrão
10. Clique no botão Alterar.

Agora que os campos foram adicionados à tabela do install.xml precisa ser salvo. Clique no link “Voltar” e “Voltar para a principal”. Uma vez na página principal clique em “Salvar” link para salvar as alterações para o install.xml. Com a ninguém install.xml criado que instala o bloco simplehtml fresco terá a montagem da tabela quando se verifique o link notificações (como discutido na Unidade 4).

## Unidade 7 – Parte B – Formulários

### Uma parte – Bloco Básico

1. Criar um simples bloco seguinte tutorial do Moodle. O bloco será nomeado “simplehtml”.
2. Criar um arquivo de linguagem e de diretório.
3. Criar uma tabela de banco de dados em que nosso bloco irá inserir informações.

### Parte B – Formulários

1. Adicionar um formulário usando o Moodle biblioteca de formulários para inserir as informações na tabela
  - \* Definir uma nova forma.
  - \* Apresentar o novo formulário.
  - \* Adicione um cabeçalho.
  - \* Adicione o controle de acesso.
  - \* Adicione registro.
  - \* Adicione elementos de formulário.
  - \* Adicione as variáveis de estado.
  - \* Finalize o formulário.
2. Processar o formulário.

### Parte C – Refinar a Interface de Usuário

1. Adicionar links para cada página do bloco.
2. Adicionar métodos para visualizar páginas.
3. Refactor método de exibição.
  - \* Adicionar título da página.
  - \* Mostrar a data
4. Exibição do texto.
5. Mostrar link para o arquivo.
6. Exibir a imagem.

### Parte D – Adicionar recursos de edição

Adicionar recursos de edição que permitem aos usuários editar dados, incluindo

1. Excluir uma página
2. Adicionar um link para o arquivo delete.php
3. Criar o arquivo delete.php
4. Limpar todos os dados associados com o bloco com o instance\_delete () função

## Parte E – definir um papel para o Bloco

1. Adicionar verificação para o bloco para a capacidade
2. Adicionar um papel para a forma simplehtml
3. Adicionar uma função para editar
4. Adicionar uma função para apagar

Noções básicas de bloqueio | Formulários | Refine the User Interface | Adicionar recursos de edição | definir um papel

Hyperlinks abrir em novas janelas do navegador.

Na parte A desta unidade, você aprendeu como criar um bloco básico com todas as funções padrão que Moodle precisa exibir o bloco com alguma configuração limitada. Você aprendeu a criar uma tabela de banco de dados usando o editor de XMLDb no Moodle, acrescentou um novo ficheiro de linguagem para o bloco.

Em seguida, você vai trabalhar na criação de um formulário para adicionar informações à tabela do banco de dados que você criou.

Adicionar um formulário para um Bloco Usando a biblioteca de formulários

Leia através de documentação do Moodle em formas de uso da biblioteca e da definição dos campos de formas biblioteca. Depois de ler sobre como criar um formulário, criar um para o bloco `block_simplehtml` que permite ao usuário:

1. Digite o texto para descrever a página html o usuário estará indo para quando clicarem em um link no bloco
2. Exibir um link para um arquivo no diretório de arquivo do curso
3. Exibir a imagem (responder a uma pergunta Sim ou Não)
4. Selecionar entre três imagens para mostrar
5. Digite o texto para a área sob a imagem html
6. Exibir uma data e hora
7. Colocar os campos criados no # 3, # 4 e # 5 (acima) em um fieldset separado
8. Adicione o campo criado em # 6 (acima) para a seção de recursos avançados de uma forma
9. Exigir # 1 (acima) para ser introduzido antes do formulário pode ser enviado.

Definir uma nova forma

O primeiro passo para criar uma forma é criar a classe de formulário que se estende `moodleform`. Crie um arquivo chamado `simplehtml_form.php` (observando as convenções de nomenclatura do Moodle, consulte o Apêndice A) e adicione o seguinte código:

```
<? Php
require_once (“$ CFG-> libdir / formslib.php”);
classe simplehtml_form estende moodleform {
definição da função () {
global $ CFG;
Mform $ = & $ this-> _form;
$ Mform-> addElement (‘cabeçalho’, ‘displayinfo, get_string (‘ textfields ‘,’ block_simplehtml ‘));
}
}
?>
```

A “definição” é uma função que será substituído e acrescenta os elementos estáticos de um formulário. Esta é a definição de classe básica para o formulário. Sempre que você quiser criar um formulário que você pode copiar o código acima e substituir “simplehtml” com o nome do formulário que você está criando. O código adiciona um cabeçalho HTML com a forma, o que

significa que a sequência de campos de texto precisa ser adicionada ao arquivo de idioma para o bloco.

Exibir um formulário

Agora que a definição do formulário básico foi criado, o próximo passo é criar o programa php que irá exibir o formulário. Criar um novo arquivo em blocos / diretório chamado simplehtml view.php. A este arquivo, adicione o seguinte código:

```
<? Php
require_once ('../../ config.php ');
require_once ('simplehtml_form.php');
$ Simplehtml = new simplehtml_form ();
$ Simplehtml-> display ();
?>
```

O código acima irá exibir e processar o formulário. Você pode testá-lo clicando no link addPage no rodapé do bloco. Como você pode ver, o uso de herança fez formulário de exibição muito simples. A classe base faz a maior parte do trabalho pesado, por isso tudo o que resta a fazer é definir uma classe de formulário que estende a classe base (moodleform) onde o display () função de membro é definido.

Embora esse código é funcional, temos ignorado muitos passos importantes. Agora vamos adicionar as peças que faltam.

Adicionar um cabeçalho

Se você visitar a página agora você vai ver uma forma muito básica não-temáticos, sem qualquer da estrutura necessária à navegação. Note-se que isso acontece porque a página de processamento do formulário, neste view.php caso, está sendo acessado diretamente através da URL e não por ser incluído em outra página. Isto significa que a infra-estrutura necessária deve ser adicionado diretamente ao script. Adicione esta usando a função print\_header, mas algumas peças de dados são necessários para a construção de navegação apropriada. Faça os ajustes a seguir para view.php:

\* Para ter acesso a US \$ CFG e \$ USER adicione o seguinte depois de require\_once ('../../ config.php ');;

CFG \$ global, \$ USER;

\* Um objeto claro povoadas é necessário que irá conter vários detalhes do curso. Desde passando o CourseID como um parâmetro, que pode ser usado para construir um objeto claro. Para fazer isso, adicione o seguinte código após require\_once ('simplehtml\_form.php'):

```
$ CourseID required_param = ('CourseID, PARAM_INT);
if (! $ curso = get_record ('curso', 'id', $ CourseID)) {
erro (get_string ('invalidcourse', 'block_simplehtml') $ CourseID.);
}
```

Note que a obtenção de um objeto válido ou gerar um erro é garantida.

\* Para exibir um cabeçalho adequado adicione o seguinte antes de \$ simplehtml-> display ();

```
print_header (strip_tags ($ site-> fullname), $ local> fullname,
"Wwwroot href=" . $CFG-> <a ' / Cursos / view.php? Id = ' . $ CourseID . ">" . $ curso-> nome
abreviado.
'</ A> ->' get_string ('formtitle', 'block_simplehtml'),",.
"<meta Name="description" content="". sumário s(strip_tags($site->)) . ">"',
verdade,"");
```

Pouco antes de \$ simplehtml-> display (); acrescentar:

```

print_header(strip_tags($ site-> fullname), $ local> fullname,
“Wwwroot href=”.$CFG-> <a ‘. / Cursos / view.php? Id =’. $ CourseID .”>’.$ curso-> nome
abreviado.
‘</ A> ->’ get_string(‘formtitle’, ‘block_simplehtml’),”,.
“<meta Name=”description” content=”‘. sumário s(strip_tags($site->)) .”>’,
verdade,”,”)

```

### Adicionar Controle de Acesso

Atualmente qualquer usuário pode visitar esta página. Agora isso não é tão grande de uma preocupação desde que o formulário não faz nada, mas sempre que os utentes tenham acesso apenas às partes do aplicativo que deveriam. Depois de carregar o objeto naturalmente, acrescentar:

```
require_login($ curso);
```

Isso irá garantir que apenas um usuário logado com acesso ao curso será capaz de utilizar o formulário simplehtml. Veja a seção Controles de acesso desta Unidade para mais detalhes.

O seu arquivo view.php agora deve ser semelhante a:

```

<? Php
require_once(‘.././ config.php ‘);
CFG $ global, $ USER;

require_once(‘simplehtml_form.php’);
$ CourseID required_param = (‘CourseID, PARAM_INT);

if (! $ curso = get_record(‘curso’, ‘id’, $ CourseID)) {
erro(get_string(‘invalidcourse’, ‘block_simplehtml’) $ CourseID.);
}

require_login($ curso);
$ Simplehtml = new simplehtml_form ();

$ Site = get_site ();
print_header(strip_tags($ site-> fullname), $ local> fullname,
“Wwwroot href=”.$CFG-> <a ‘. / Cursos / view.php? Id =’. $ CourseID .”>’.$ curso-> nome
abreviado.
‘</ A> ->’ get_string(‘formtitle’, ‘block_simplehtml’),”,.
“<meta Name=”description” content=”‘. sumário s(strip_tags($site->)) .”>’,
verdade,”,”);
$ Simplehtml-> display ();
print_footer ();

?>

```

### Adicionar Logging

Moodle usa o add\_to\_log function () para registrar informações sobre as ações do usuário. Adicione esta linha de código após a instrução if contendo insert\_record () em simplehtml.php:

```

add_to_log($ id, ‘block_simplehtml’, ‘inserir página’, $ CFG-> wwwroot. “blocos / simplehtml /
view.php? CourseID = ‘. $ CourseID.’ & blockid = ‘. $ blockid,
”, 0), $ USER-> id;

```

O primeiro parâmetro enviado para add\_to\_log é o id do curso, que é passado através da url ou formulário para simplehtml.php. O segundo parâmetro é o nome do bloco ou módulo. O parâmetro terceiro é a ação, que normalmente é algo simples. O quarto parâmetro é a url da página de referência. O quinto parâmetro é uma informação adicional se a sequência que representa a ação no terceiro parâmetro não é suficientemente descritivo. O sexto parâmetro é o id do módulo do curso. O parâmetro final é o ID de usuário.

## Formulário de Controle de Estado

Agora, para adicionar algum controle estatal básica forma: O primeiro passo é configurar as ações apropriadas com base no estado formulário, respondendo por três cenários:

1. para mostrar um formulário pela primeira vez
2. para processar dados de formulário enviado
3. o usuário cancelou o envio do formulário

Para implementar esses três cenários ajustar o código um pouco. A exibição de formulário existente e código de cabeçalho será usado como o primeiro código de tempo de exibição, e uma instrução if adicionado após instanciar o objeto `simplehtml_form` irá lidar com a lógica do fluxo da aplicação:

```
if ($ simplehtml-> is_cancelled ()) {  
  // Cancelou formas redirecionar para a página principal curso  
  redirect (" $ CFG-> wwwroot / cursos / view.php id $ id =?");  
} Else if ($ fromform = $ simplehtml-> get_data ()) {  
  // Precisamos adicionar código para agir em adequadamente e armazenar os dados apresentados  
  redirect (" $ CFG-> wwwroot / curso / view.php id = $ CourseID?");  
Else {  
  // Form não validar ou esta é a primeira exibição  
  $ Site = get_site ();  
  print_header (strip_tags ($ site-> fullname), $ local> fullname,  
  "Wwwroot href=" . $CFG-> <a ' . / Cursos / view.php? Id =' . $ CourseID . "'>' . $ curso-> nome  
  abreviado.  
  '</ A> ->' get_string ('formtitle', 'block_simplehtml'),",  
  "<meta Name='description' content=' . sumário s(strip_tags($site->)) . '>'",  
  verdade,"");  
  $ Simplehtml-> display ();  
  print_footer ();  
}
```

Observe que o cabeçalho existente e código de exibição forma foram movidos para o terceiro ramo da condicional. Além disso, não há nenhum código para lidar com processamento de formulários. Esse código será adicionado em breve.

Etapas finais

Revisar o progresso até agora:

1. `simplehtml_form.php` é criado, que define a classe usada mais tarde para mostrar o nosso formulário.
2. `view.php` é criado, que:
  - \* Base de cargas moodle API necessária e qualquer módulos de terceiros ou arquivos não-base API
  - \* Carrega o objeto curso necessário e globals
  - \* Realiza controle de acesso necessário
  - \* Carrega o nosso formulário e execução ramos adequadamente com base em nosso estado formulário

Assim, a classe de formulário é definido e o formulário é exibido em um arquivo separado. Adicionando elementos para o formulário e terminando o tratamento de nosso formulário irá envolvê-lo!

Adicionar elementos de formulário

Atualmente não há elementos de formulário que pode enviar dados. Edição da definição da classe em forma `simplehtml_form.php` vai mudar isso adicionando um vasto conjunto de elementos de formulário para o formulário para fins de demonstração somente. Isso vai dar um exemplo de como adicionar e processo de elementos de formulário vários.

Para adicionar o link campo de texto eo título displaytext área de HTML para o formulário, cole o seguinte texto após a criação do objeto de formulário no arquivo simplehtml\_form.php:

```
// Adiciona grupo para áreas de texto
$ Mform-> addElement ('cabeçalho', 'displayinfo, get_string (' textfields ', ' block_simplehtml '));

// Adiciona elemento título da página
$ Mform-> addElement ('text', 'pagetitle, get_string (' pagetitle ', ' block_simplehtml '));
$ Mform-> addRule ('pagetitle', null, 'required', null, "cliente");

// Adiciona campo de texto de exibição
$ Mform-> addElement ('htmleditor', 'displaytext, get_string (' displayedhtml ', ' block_simplehtml '));
$ Mform-> settype ('displaytexttext, PARAM_RAW);
$ Mform-> addRule ('displaytext', null, 'required', null, "cliente");
```

A página irá agora mostrar:

adicionando os campos do formulário para a forma

Para maiores detalhes sobre addElement () analisar a documentação sobre moodle addElement.

Também addRule revisão () na documentação do Moodle e no site da PEAR. PEAR é uma biblioteca de processamento de forma que incluiu moodle usa para processamento de formulários.

Agora adicionar elementos de formulário adicionais para simplehtml\_form.php:

File Upload

Adicione o campo arquivo para o formulário após os elementos existentes:

```
// Adiciona a seleção filename
$ Mform-> addElement ('choosecoursefile', 'filename', get_string ('displayfile',
'block_simplehtml'), array ('CourseID' => $ COURSE-> id));
```

Este campo permite que o usuário escolha um arquivo no diretório de arquivos do curso. Observe que o nome do elemento em forma é o mesmo nome do campo no banco de dados. Isto tornará mais fácil de inserir e atualizar os registros do banco de dados posteriormente. Observe que o código usa \$ COURSE-> id: isso significa que a variável \$ COURSE precisa ser adicionado à linha de variáveis globais próxima a US \$ CFG.

Este próximo conjunto de widgets permite a exibição de uma imagem opcional o usuário seleciona através de um botão de rádio, e adiciona um texto alternativo que é exibido com a imagem. Você provavelmente fazer isso diretamente através do editor de HTML, mas é adicionado ao formulário para ilustração.

Cabeçalho fieldset

```
// Adiciona cabeçalho picturefields
$ Mform-> addElement ('cabeçalho', 'pictureinfo, get_string (' picturefields ', ' block_simplehtml '));
```

Yes / No Select

```
// Adiciona imagem de exibição sim / não opção
$ Mform-> addElement ('selectyesno', 'displaypicture, get_string (' displaypicture ',
'block_simplehtml '));
$ Mform-> setDefault ('displaypicture', 1);
```

Observe o valor padrão para a imagens de exibição é definido como "sim".

Selecione Rádio

Para controlar as opções disponíveis, criar uma função que irá retornar a lista de opções disponíveis como uma matriz chave por um número inteiro. Este é o preferido por várias razões:

\* Mantém o banco de dados livre de tipos de campo enum, fornecendo importantes dados por

números inteiros. Este é o motor de banco de dados independentes.

- \* Isso torna mais fácil para adicionar opções adicionais em uma data posterior, alterando a nossa função

- \* Evita o uso de variáveis globais.

Criar um lib.php arquivo e coloque o seguinte código para ele que vai definir as opções de imagem via rádio selecionar.

```
<? Php
/ **
* Funções de biblioteca para o bloco simplehtml
** /
block_simplehtml_images function () {
global $ CFG;
return array ('src="' <img. wwwroot $CFG->.' / blocks/simplehtml/pix/picture0.jpg "alt ="
"get_string ('vermelho', 'block_simplehtml').">',
"<img Src="' . Wwwroot $CFG->. ' / Blocks/simplehtml/pix/picture1.jpg" alt = "' . Get_string ('
block_simplehtml azul ', '>',
"<img Src="' . Wwwroot $CFG->. ' / Blocks/simplehtml/pix/picture2.jpg" alt = "' . Get_string ('
verde ', 'block_simplehtml').">');
}
?>
```

Agora inclua o arquivo acima em todos os arquivos que precisam usar a função, por enquanto isso é apenas simplehtml\_form.php. Adicione o seguinte depois de require\_once (“\$ CFG-> libdir / formslib.php”); em simplehtml\_form.php

```
require_once ($ CFG-> dirroot ' / blocos / simplehtml / lib.php.);
```

Adicionar as fotos acima referenciado para o diretório apropriado. Crie um diretório chamado pix em seu diretório bloco simplehtml, e extraia o arquivo a seguir lá. Agora adicione o rádio selecionar para a definição do formulário (simplehtml\_form.php)

```
// Adicionar botões imagem seletor de rádio
$ Imagens = block_simplehtml_images ();
$ Radioarray = array ();
for ($ i = 0; $ i <count ($ imagens); $ i ++ ) {
$ Radioarray [] = & $ mform-> createElement ('rádio', 'imagem', '', $ imagens [$ i], $ i);
}

$ Mform-> addgroup ($ radioarray, 'radioar, get_string (' pictureselect ', 'block_simplehtml '),
array (' ', false);
```

Um grupo também é adicionado para capturar múltiplos campos do formulário em uma linha. Outra abordagem seria apenas exibir cada elemento de rádio em sua própria linha.

Alt Text Input

Finalmente, para envolver-up seleção exibir a imagem, permitem ao usuário escolher o texto alternativo que será exibido com a imagem selecionada.

```
// Adiciona campo de descrição
$ Atributos = array ('size' => '50 ', 'maxlength' => '100');
$ Mform-> addElement ('text', 'description', get_string ('picturedesc', 'block_simplehtml'), $
atributos); $ mform-> settype ('description', PARAM_TEXT);
$ Mform-> settype ('description', PARAM_TEXT);
```

Observe que há uma variável de atributos adicionados ao campo de texto para limitar o tamanho e comprimento. O setAdvanced () função adiciona opções avançadas a um formulário. Para usar esta

função em passar o nome do elemento do formulário, o grupo de elementos, ou cabeçalho. O “Advanced Options” aparece no canto superior direito do fieldset. Neste exemplo, existe um único elemento, mas depois você verá um elemento dentro de um cabeçalho. Saiba mais sobre setAdvanced. O “Show / Hide Advanced” botão não será exibido até que a adição de botões de ação para a forma durante a próxima etapa deste tutorial.

Selector data Hora

Agora adicione o campo de data como um elemento opcional:

```
// Adicionar agrupamento opcional
$ Mform-> addElement ('cabeçalho', 'opcional', get_string ('opcional', 'forma'), null, false);

// Adiciona seletor date_time na área opcional
$ Mform-> addElement ('date_time_selector', 'DisplayDate', get_string (' DisplayDate ', '
block_simplehtml '), array (' opcional '=> true));
$ Mform-> setAdvanced ('opcionais');
```

O código acima adiciona um cabeçalho do formulário e define-a ocultar a data / seletor de tempo dentro do opcional controles (avançado) formulário.

Adicionar variáveis de estado e Finalize o Formulário

Adicionando botões Form

Formas mais moodle tem um conjunto básico de botões. “Submit” e “Cancel” são adicionados chamando o moodleform add\_action\_buttons membro function (). Adicione o seguinte no final do simplehtml\_form.php

```
$ This-> add_action_buttons ();
```

Observe quando add\_action\_buttons chamando (), \$ this é usado ao invés de \$ mform porque add\_action\_buttons () é definido na classe base moodleform e seria escopo fora por US \$ mform.

Adicionar Variáveis Estado

Às vezes, os formulários devem manter o controle de dados adicionais que não se destina a ser visível aos usuários. Estes campos devem ser colocados em elementos ocultos. Para este blockid forma eo CourseID são necessários. Antes da chamada para add\_action\_buttons () acrescentar:

```
$ Mform-> addElement ('escondido', 'blockid');
$ Mform-> addElement ('escondido', 'CourseID');
```

Estes elementos terão de ser preenchida com a exibição de código porque estes serão passados através da URL inicialmente. Antes da função de exibição em view.php acrescentar:

```
$ Toform ['blockid'] = $ blockid;
$ Toform ['CourseID'] = $ CourseID;
$ Simplehtml-> set_data ($ toform);
```

Adicionar \$ blockid como um parâmetro obrigatório no formulário de exibição view.php arquivo.

Adicione este código após \$ CourseID required\_param = ('CourseID, PARAM\_INT);;

```
$ Blockid required_param = ('blockid, PARAM_INT);
```

Uma nota final: strings de linguagem não foram adicionadas à linguagem lang/en\_utf8/block\_simplehtml.php arquivo. Adicionar as entradas apropriadas para o arquivo de idioma. Lembre-se a presença de uma cadeia cercada por dois colchetes (por exemplo, SomeString) indica que a corda está faltando uma definição de linguagem. Depois de adicionar strings seu idioma seu formulário deve ser algo como:

imagem de exemplo da forma moodle

Formulário de Dados de processo

Notas importantes



\* Use `add_to_log ()` com moderação. Cada bloco, módulo e código PHP será adicionando outros dados para a tabela de log. Quanto mais informações na tabela de log, o menos útil se torna mais lento e as consultas de banco de dados sobre a mesa.

\* Sempre use `add_to_log ()` quando um usuário executa ações como adicionar, editar e excluir. Ações fora de adicionar, editar e apagar deve ser usado com moderação. Por exemplo, pode ser útil para um instrutor para saber quando (ou se) o aluno acessou determinado material do curso.

\* Um dos mais importantes Moodle within aspectos administrativos está registrando as ações do usuário. Isso ajuda os administradores do site diagnosticar problemas, apanhar batoteiros, e, em geral descobrir o que um usuário fez para chegar onde eles tinham um problema. Isso também pode ser de valor inestimável para os desenvolvedores para determinar onde no seu código alguém foi capaz de chegar até antes de terem um problema. Claro, os logs também são críticos para os professores para monitorar as atividades do estudante em um curso.

Depois de criar o formulário, código apropriado deve ser adicionado ao processo os dados do formulário apresentado. Fazer isso adicionando código para `view.php`. Normalmente durante o processamento de dados do formulário é útil para remover parte do processamento final de passos para que seu script essencialmente termina a execução e não está redirecionado para a página do curso ou onde quer que faz sentido. Isso é extremamente útil durante a depuração. O código de processamento será adicionado a este bloco de código em `view.php`:

```
else if ($ fromform = $ simplehtml-> get_data ()) {  
// Precisamos adicionar código para agir em adequadamente e armazenar os dados apresentados  
redirect (“$ CFG-> wwwroot / curso / view.php id = $ CourseID?”);  
}
```

Por agora comentar o redirecionamento e adicione o seguinte sobre ele

```
print_object ($ fromform);
```

`print_object ()` é uma função útil moodle que imprime os dados de tipos de dados mistos mostrando as chaves e dados para arrays e objetos. Agora, visite o “Add Page” link para o bloco e apresentar alguns dados do formulário. Você deverá ver algo semelhante a:

Objeto `stdClass`

```
(  
[MAX_FILE_SIZE] => 2097152  
[Pagetitle] => Esta é a minha primeira página simplehtml  
[Displaytext] => Lorem ipsum dolor sit amet ...  
[Arquivo] => blockimage1.gif  
[Displaypicture] => 1  
[Foto] => 0  
[Description] => O vermelho é uma cor grande  
[DisplayDate] => 1222722600  
[Mform_showadvanced_last] => 1  
[Blockid] => 13  
[CourseID] => 2  
[Submitbutton] => Salvar alterações  
)
```

Os valores entre parênteses são as chaves e depois a parte `=>` é o de dados para que a chave particular. Desde o envio do formulário é verificado como está funcionando como esperado, salvar os dados do formulário apresentado. A maneira mais direta para salvar é com uma chamada para `insert_record ()`. Esta função recebe o nome da tabela, ‘`block_simplehtml`’ no nosso caso, eo objeto a ser inserido. O objeto deve ter chaves que mapeiam um-para-um com os nomes de coluna da tabela. Nomes de formulário escolhendo elemento que mapeiam diretamente para as colunas de banco de dados economiza um pouco de código. Salvar os dados do formulário apresentado com o

seguinte código colado acima da out comentou redirect:

```
if (! insert_record ('block_simplehtml', $ fromform)) {  
erro (get_string ('inserterror', 'block_simplehtml'));  
}
```

Observe que após a verificação de uma chamada para insert\_record o status de OK é retornado e, se não um erro é produzido. Agora comentar o print\_object () e não-comentar a chamada para redirect (). Finalmente, teste para ver se o envio do formulário e salvar funciona corretamente, apresentando alguns dados utilizando o formulário. Você pode analisar que os dados foram armazenados na tabela apropriada visualizando a tabela block\_simplehtml via phpmyadmin. Você deverá ver algo semelhante à esta:

imagem de registro no phpmyadmin

Observação Se você receber um erro, verifique se sua tabela existe no banco de dados. Lembre-se de atualização do código não foi adicionado ao instalar o banco de dados tabela após o bloco já foi instalado. Ao desenvolver dentro de um ambiente localizado carregar o banco de dados por desinstalar o módulo e, em seguida, reinstalá-lo. Em circunstâncias normais, onde o bloco ou módulo está em uso em produção você precisa fazer isso com o código de atualização correta.

## Unidade 7 – Parte C – Refinar a Interface de Usuário

### Uma parte – Bloco Básico

1. Criar um simples bloco seguinte tutorial do Moodle. O bloco será nomeado “simplehtml”.
2. Criar um arquivo de linguagem e de diretório.
3. Criar uma tabela de banco de dados em que nosso bloco irá inserir informações.

### Parte B – Formulários

1. Adicionar um formulário usando o Moodle biblioteca de formulários para inserir as informações na tabela
  - \* Definir uma nova forma.
  - \* Apresentar o novo formulário.
  - \* Adicione um cabeçalho.
  - \* Adicione o controle de acesso.
  - \* Adicione registro.
  - \* Adicione elementos de formulário.
  - \* Adicione as variáveis de estado.
  - \* Finalize o formulário.
2. Processar o formulário.

### Parte C – Refinar a Interface de Usuário

1. Adicionar links para cada página do bloco.
2. Adicionar métodos para visualizar páginas.
3. Refactor método de exibição.
  - \* Adicionar título da página.
  - \* Mostrar a data
4. Exibição do texto.
5. Mostrar link para o arquivo.
6. Exibir a imagem.

### Parte D – Adicionar recursos de edição

Adicionar recursos de edição que permitem aos usuários editar dados, incluindo

1. Excluir uma página
2. Adicionar um link para o arquivo delete.php

3. Criar o arquivo delete.php
4. Limpar todos os dados associados com o bloco com o instance\_delete () função

Parte E – definir um papel para o Bloco

1. Adicionar verificação para o bloco para a capacidade
2. Adicionar um papel para a forma simplehtml
3. Adicionar uma função para editar
4. Adicionar uma função para apagar

Noções básicas de bloqueio | Formulários | Refine the User Interface | Adicionar recursos de edição | definir um papel

Após o sucesso de enviar dados ao banco de dados através do formulário, é chegada a hora de começar a exibir os dados que os usuários possam interagir com os dados e ver o que tem sido submetido. Ambas as etapas serão contidos simplehtml / block\_simplehtml.php.

Adicionar links para cada página do Bloco

Para recuperar os registros relevantes chamada get\_records (), que aceita três parâmetros::

1. o nome da tabela
2. o nome da coluna ou campo a ser consultado
3. o valor do campo consultado queremos jogo (note o uso de \$ this-> exemplo-> id, que é o mesmo que blockid neste âmbito)

Adicione o seguinte código após \$ this-> content-> text = \$ this-> config-> Texto:

```
if ($ simplehtmlpages get_records = ('block_simplehtml', 'blockid', $ this-> exemplo-> id)) {
$ This-> conteúdo> texto .= '<ul class="block-simplehtml-pagelist">';
foreach ($ simplehtmlpages quanto $ simplehtmlpage) {
$ This-> conteúdo> texto .= '<li> <a href="' . $CFG-> wwwroot .
'/ Blocos simplehtml / / view.php? Id ='. $ Simplehtmlpage-> id .
'& CourseID =' $ COURSE-> id .'">' . $ simplehtmlpage-> pagetitle '</ a> </ li>' ..;
}
$ This-> conteúdo> texto .= '</ ul>';
}
```

Adicionar Métodos para ver páginas

Reveja os links criados anteriormente para ver que o view.php arquivo é necessário que:

1. Verifique se há qualquer variável necessária; id (o id referências página simplehtml), CourseID
2. Verificar a CourseID dado é um curso válido
3. Verifique se o usuário está logado
4. Verifique thatte usuário tem as permissões adequadas para ver qualquer thescript saída pode gerar
5. Recuperar o registro de banco de dados para a página solicitada simplehtml
6. Imprimir o cabeçalho da página
7. Exibir as informações para a página requisitada
8. Imprimir o rodapé da página

Muitas dessas medidas já foram concluídas. Tente trabalhar com a lista acima, pulando o “informações de exibição para a página solicitada”. Seu código deve ser semelhante à:

```
<? Php
// Include moodle API e documentos suplementares / API
require_once ('../ config.php ');
require_once ('simplehtml_form.php');
// Declarar qualquer globals precisamos usar
```

```

CFG $ global, $ USER;

// Verificar todas as variáveis necessárias
$ Id = required_param ('id', PARAM_INT);
$ CourseID required_param = ('CourseID', PARAM_INT);

// Garantir que tenhamos uma CourseID válido e pode carregar o objeto curso associado
if (! $ curso = get_record ('curso', 'id', $ CourseID)) {
erro (get_string ('invalidcourse', 'block_simplehtml', $ CourseID));
}

// Assegura que o usuário tem acesso a este curso
require_login ($ curso);

// Assegura que o usuário tem as permissões adequadas para acessar esta área
require_capability ('bloco / simplehtml: viewpages, get_context_instance (CONTEXT_COURSE, $
CourseID));

// Assegura que temos um ID de página válido simplehtml e pode carregar a página associada
if (! $ simplehtmlpage get_record = ('block_simplehtml', 'id', $ id)) {
erro (get_string ('nopage', 'block_simplehtml', $ id));
}

// Imprimir o cabeçalho e os dados associados
$ Site = get_site ();
print_header (strip_tags ($ site-> fullname), $ local> fullname,
"Wwwroot href=".$CFG-> <a ' / Cursos / view.php? Id =' $ CourseID .'">'.$ curso-> nome
abreviado.
'</ A> ->'.$ simplehtmlpage-> pagetitle",
"<meta Name="description" content="". s(strip_tags($site-> resumo)) .'">', verdade,"");

/ Informação da página / display

// Imprimir o rodapé
print_footer ();
?>

```

Note o uso de um terceiro parâmetro com `get_string ()`. O terceiro parâmetro permite que o sistema Moodle de tradução para substituir variáveis em locais apropriados depois de ser traduzido. Consideremos um exemplo do código acima.

```
erro (get_string ('invalidcourse', 'block_simplehtml', $ CourseID));
```

Este `get_string` chamadas () em busca de 'invalidcourse' no arquivo de idioma 'block\_simplehtml' e, finalmente, ele passa em \$ CourseID. Em seguida, no arquivo de idioma para o bloco esta cadeia terá acesso ao valor passado através do Parâmetro de terceiros como \$ a.

```
$ String ['invalidcourse'] = 'O $ CourseID não existe';
```

### Refactor Método de exibição

Para evitar a duplicação de código e tornar mais fácil a reutilização esta funcionalidade criar uma função em `lib.php` para lidar com a exibição da página. Uma página `simplehtml` pré-carregados é passado como um único parâmetro, e um parâmetro opcional irá controlar se os dados são devolvidos ou diretamente impressos. A maioria das funções do Moodle que imprimir as informações pode ser passado um valor verdadeiro como o último parâmetro. Este é o valor de retorno da função, que conta a função para retornar o HTML e não imprimir ou eco-la para o usuário. Para o desenvolvimento, defina o valor de retorno para false para que os dados são impressos.

```
block_simplehtml_print_page função ($ simplehtml, $ retorno = FALSE) {}
```

ponto de exclamação

Dica importante

A função usa `block_simplehtml_print_page` `block_simplehtml` no início do nome da função para evitar conflitos de namespace. Conflitos de namespace são importantes a considerar ao nomear variáveis e funções em Moodle Moodle beause tem muitos ganchos que chamam funções diferentes em momentos diferentes. É geralmente considerada uma boa prática de prefaciar um nome variable ou função com o nome do bloco ou módulo para evitar problemas com conflitos de namespace.

Agora considere o que a função precisa fazer a fim de exibir todas as informações necessárias a partir de nossa página.

Título da página

Imprima esta fora como um cabeçalho centrado para a página HTML

Texto de exibição

Imprima esta fora em qualquer formato que o usuário especificado, centrado em uma caixa na página

Nome do arquivo

Exibir um link para o arquivo no diretório de arquivo do curso para que o usuário tem a opção de baixá-lo. Este será centrada na caixa mesmo que o texto de exibição.

Imagem de Exibição

Exibir a imagem com a descrição no lado direito da imagem em uma caixa, se o usuário escolheu para a foto para mostrar

Exibição de data

Exibir a data sob o título de página em uma fonte menor

Adicionar Título da página

Para exibir a página moodle `print_heading` use `título ()` função. Adicione o seguinte ao `block_simplehtml_print_page ()` em `lib.php`

```
$ Output = print_heading ($ simplehtml-> pagetitle, "centro", 2, 'main', TRUE);
```

Os parâmetros em ordem são 'texto', 'text-alinhamento', 'tamanho do cabeçalho', 'css classe', finalmente, se para imprimir diretamente a saída ou devolvê-lo.

Exibir o Date

Os requisitos para a chamada data de exibição para exibir a data ni uma fonte menor sob o título da página. Em primeiro lugar, mostrar a data. Moodle tem uma função `userdate ()` que tem várias vantagens sobre PHP `date ()` função. Ele exibe a data no formato preferido do usuário e ajusta a data para qualquer diferença de fuso horário.

```
if ($ simplehtml-> DisplayDate) {  
$ Output .= userdate ($ simplehtml-> DisplayDate);  
}
```

Note que desde que o usuário tinha a opção de escolher uma data ou não que você deve verificar para garantir que ele foi criado antes de tentar a saída dele. Dê uma olhada na página exibida: clique em um dos links a página do bloco. Não há saída. Adicione um pouco de código adicional que lida com o valor de US \$ retorno. Adicione o seguinte no final da definição da função:

```
if ($ retorno) {  
return $ resultado;  
}  
else {  
print $ saída;  
}
```

Agora tente atualizar sua página. Você deve ver algo como:

Como mencionado anteriormente, os requisitos para o campo da data fosse centrada sob o título da página e menores. Isto pode ser controlado com uma folha de estilo.

Primeiro, adicione uma tag div em torno da data, e dar-lhe uma classe de alvo com CSS mais tarde. Alterar a linha de saída para data:

```
$ Output .= '<div>' userdate ($ simplehtml-> DisplayDate ).'</ div> ';
```

Há duas opções para adicionar a folha de estilo:

1. Adicione o CSS para arquivos CSS do tema no diretório do tema
2. Crie um arquivo estilos.php no bloco e adicionar o CSS lá

Escolhendo a segunda opção, de adicionar um arquivo estilos.php, permite mais flexibilidade e evita a necessidade de criar várias entradas se o site está usando mais de um tema. Criar estilos.php e adicione este CSS:

```
. Simplehtml.displaydate {  
font-size: 0,8 em;  
text-align: center;  
}
```

Agora atualize a página ea data deve ser exibido corretamente.

Texto de exibição

Agora o texto de exibição pode ser adicionada a uma caixa.

Dentro moodle existem duas maneiras de exibir uma caixa:

1. print\_box () e
2. uma combinação de print\_box\_start () e print\_box\_end ().

A saída é o mesmo; o que você escolher é principalmente uma questão de preferência. Para adicionar duas peças diferentes de dados para a caixa, display de texto e no link do arquivo, o código de exemplo usa print\_box\_start () e print\_box\_end () para facilitar a leitura.

```
$ Output .= print_box_start ('generalbox',", TRUE);  
$ Output .= clean_text ($ simplehtml-> displaytext);  
$ Output .= print_box_end (TRUE)
```

Mostrar Vincular ao Arquivo

Para gerar uma URL de arquivo chamar o get\_file\_url function (), mas a função é definida em lib / filelib.php e não é uma parte do moodle núcleo API. Assim, garantir que você tenha acesso a este arquivo chamando include\_once (). Adicione o seguinte próximo ao topo da função:

```
CFG $ global, CURSO $;  
include_once ($ CFG-> dirroot ' / lib / filelib.php.);
```

Agora adicione as seguintes linhas antes da chamada para print\_box\_end ()

```
$ Fileurl get_file_url = ($ COURSE-> id .' / '$ simplehtml-> filename);  
$ Output .= '<br /> href="' . $fileurl .'">' get_string ('viewfile', 'block_simplehtml').'</ a>' .;
```

Observe que o fornecimento de mais do que apenas o nome do arquivo do arquivo para criar uma URL apropriada é necessária. O ID de curso para o curso em questão também deve ser fornecido.

Exibir a imagem

Seletivamente exibir a imagem em seu texto de descrição localizado à direita do mesmo:

```
if ($ simplehtml-> displaypicture) {  
$ Imagens = block_simplehtml_images ();  
$ Output .= print_box_start ('generalbox',", TRUE);
```

```
$ Output .= $ imagens [$ simplehtml-> foto];  
$ Output .= $ simplehtml-> descrição;  
$ Output .= print_box_end (TRUE);  
}
```

A página deve agora olhar como:

## Unidade 7 – Parte D – Adicionar recursos de edição

### Uma parte – Bloco Básico

1. Criar um simples bloco seguinte tutorial do Moodle. O bloco será nomeado “simplehtml”.
2. Criar um arquivo de linguagem e de diretório.
3. Criar uma tabela de banco de dados em que nosso bloco irá inserir informações.

### Parte B – Formulários

1. Adicionar um formulário usando o Moodle biblioteca de formulários para inserir as informações na tabela
  - \* Definir uma nova forma.
  - \* Apresentar o novo formulário.
  - \* Adicione um cabeçalho.
  - \* Adicione o controle de acesso.
  - \* Adicione registro.
  - \* Adicione elementos de formulário.
  - \* Adicione as variáveis de estado.
  - \* Finalize o formulário.
2. Processar o formulário.

### Parte C – Refinar a Interface de Usuário

1. Adicionar links para cada página do bloco.
2. Adicionar métodos para visualizar páginas.
3. Refactor método de exibição.
  - \* Adicionar título da página.
  - \* Mostrar a data
4. Exibição do texto.
5. Mostrar link para o arquivo.
6. Exibir a imagem.

### Parte D – Adicionar recursos de edição

Adicionar recursos de edição que permitem aos usuários editar dados, incluindo

1. Excluir uma página
2. Adicionar um link para o arquivo delete.php
3. Criar o arquivo delete.php
4. Limpar todos os dados associados com o bloco com o instance\_delete () função

### Parte E – definir um papel para o Bloco

1. Adicionar verificação para o bloco para a capacidade
2. Adicionar um papel para a forma simplehtml
3. Adicionar uma função para editar
4. Adicionar uma função para apagar

Noções básicas de bloqueio | Formulários | Refine the User Interface | Adicionar recursos de edição | definir um papel

Alunos e professores podem ver os links eo professor pode criar links, mas o que acontece se eles

cometem um erro? Não há nenhuma maneira de editar qualquer coisa que tenha sido previamente inscritas. Bons desenvolvedores querem que os seus blocos a serem utilizados, o que significa que o bloco precisa incluir uma forma de o usuário editar os dados existentes. A maneira de editar o que um usuário entrou deve ser adicionado ao bloco simplehtml. O recurso de edição irá usar uma imagem de linkar a edit.php em vez de texto. Esta imagem só aparece quando o instrutor se transforma em edição.

Fotos Moodle lojas e ícones no diretório pix no diretório moodle base. Você também pode usar imagens que são personalizadas com um tema. Estes são armazenados dentro do diretório do tema pix. Para este exemplo, use os ícones no diretório pix para moodle. A imagem deve ser encapsulado em um link HTML para simplehtml.php. A url deve passar o id instância do bloco, id bloco, e id curso para simplehtml.php.

Antes de exibir a edição acrescentou imagem e link de verificação que a edição foi ativado (verificando se o usuário tem as permissões adequadas é abordado na última seção, Controles de Acesso). O isediting function () é usado para determinar se o usuário tiver a edição activada para o curso. O isediting () função é passado o pageid para a instância do bloco, que na maioria dos casos é o mesmo que CourseID. Editar o arquivo block\_simplehtml.php, como segue:

```
if ($ simplehtmlpages get_records = ('block_simplehtml', 'blockid', $ this-> exemplo-> id)) {
$ This-> conteúdo> texto .= '<ul class="block-simplehtml-pagelist">';
foreach ($ simplehtmlpages quanto $ simplehtmlpage) {
if ($ canmanage) {
$ Editar = 'wwwroot href="'. $CFG-> <a' . / Blocos simplehtml / / view.php? Id = '
. $ Simplehtmlpage-> id. '& Blockid ='. $ This-> exemplo-> id. '& CourseID ='.
$ COURSE-> id .'"> </ a>';
Else {}
$ Editar ='';
}
$ This-> conteúdo> texto .= '<li> <a href="'. $CFG-> wwwroot.
'/ Blocos simplehtml / / view.php? Id ='. $ Simplehtmlpage-> id.
'& CourseID =' $ COURSE-> id .'">'. $ simplehtmlpage-> pagetitle '</ a>' $ edit '</ li>' ....;
}
$ This-> conteúdo> texto .= '</ ul>';
}
```

Repare que há um link para o view.php mesmo arquivo que lidou com a página adicionando. Fazer alguns ajustes para view.php para que as páginas também podem ser atualizados.

Em primeiro lugar, modificar a forma de ter um novo campo oculto chamado id para armazenar o id do registro no banco de dados para saber qual página para atualização. Adicione o seguinte ao simplehtml\_form.php:

```
$ Mform-> addElement ('escondido', 'id', '0');
```

Depois, adicione um parâmetro opcional para a forma de rastrear o id no caso de você atualizar uma página existente. Adicione o seguinte código para view.php

```
$ Id = optional_param ('id', 0, PARAM_INT);
```

Em terceiro lugar, ajustar a exibição pela primeira vez ou não forma parte exibir validação de view.php para carregar os dados necessários quando id \$ está presente:

```
// Form não validar ou esta é a primeira exibição
if ($ id = 0) {
if (!$ toform get_record = ('block_simplehtml', 'id', $ id)) {
erro (get_string ('nopage', 'block_simplehtml', $ id));
```



```

}
Else {}
$ Toform = new stdClass;
}
$ Toform-> blockid = $ blockid;
$ Toform-> CourseID = $ CourseID;
$ Site = get_site ();
print_header (strip_tags ($ site-> fullname), $ local-> fullname,
"Wwwroot href=" . $CFG-> <a ' / Cursos / view.php? Id =' . $ CourseID . "'>' . $ curso-> nome
abreviado.
'</ A> ->' get_string ('formtitle', 'block_simplehtml'),",",.
"<meta Name="description" content=" . s(strip_tags($site-> resumo)) . "'>' , verdade,",",");
$ Simplehtml-> set_data ($ toform);
$ Simplehtml-> display ();
print_footer ();

```

Observe que a manipulação da variável \$ toform foi ajustado. Agora é um objeto ao invés de um array. Isso é para dar consistência, independentemente de nós estamos adicionando uma nova página ou atualizar um já existente.

Por fim, altere o código para condicionalmente atualizar ou inserir um novo registro. No bloco do meio condicional (elseif) ajustar o seguinte para atualizar os registros quando o campo id é diferente de zero

```

// Adiciona o código para agir em adequadamente e armazenar os dados apresentados
if ($ fromform-> id = 0) {
if (! update_record ('block_simplehtml', $ fromform)) {
erro (get_string ('updateError', 'block_simplehtml'));
}
add_to_log ($ blockid, 'block_simplehtml', 'page update', $ CFG-> wwwroot.
"Blocos simplehtml // view.php? Blockid = ' . $ Blockid.' & CourseID ' .
$ CourseID,", $ blockid, $ USER-> id);
Else {}
if (! insert_record ('block_simplehtml', $ fromform)) {
erro (get_string ('inserterror', 'block_simplehtml'));
}
add_to_log ($ id, 'block_simplehtml', 'inserir página', $ CFG-> wwwroot. "blocos / simplehtml /
view.php? CourseID = ' . $ CourseID.' & blockid = ' . $ blockid,
", 0), $ USER-> id;
}
redirect (" $ CFG-> wwwroot / curso / view.php id = $ CourseID?");

```

### Supressão página

Depois de ser capaz de adicionar uma página e editar uma página, o que mais alguém precisa fazer com as nossas páginas html? Apagar! Exclusão é uma operação simples, requerendo apenas alguns passos simples::

1. Adicionar um link para um arquivo novo delete.php
2. Criar o arquivo delete.php
3. Imprimir um formulário de verificação para a tela caso o usuário cometeu um erro
4. Use delete\_record () para excluir um registro da tabela simplehtml dado um id e uma confirmação.

Adicione um link para delete.php

Comece editando block\_simplehtml.php e criar um link logo após o link de edição.

```

“Wwwroot href=”$.CFG-> <a. ‘$ Delete = / blocos / simplehtml / delete.php? Id =’.
$ Simplehtmlpage-> id. ‘& CourseID =’. $ COURSE-> id .”>< img src = “.
$ CFG-> pixpath. ‘/ T / delete.gif “alt =” “.
get_string (‘DeletePage’, ‘block_simplehtml’).” /> </ a>’;

```

Lembre-se de definir o links para uma string em branco se o usuário não tem as permissões corretas:

```

if ($ canmanage) {
$ Editar = ‘wwwroot href=”$.CFG-> <a. / Blocos simplehtml / / view.php? Id = ‘
. $ Simplehtmlpage-> id. ‘& Blockid =’. $ This-> exemplo-> id. ‘& CourseID =’.
$ COURSE-> id .”>< img src =”$.CFG-> pixpath. ‘/ T / edit.gif “alt =” “.
get_string (‘editpage’, ‘block_simplehtml’).” /> </ a>’;

“Wwwroot href=”$.CFG-> <a. ‘$ Delete = / blocos / simplehtml / delete.php? Id =’.
$ Simplehtmlpage-> id. ‘& CourseID =’. $ COURSE-> id .”>< img src = “.
$ CFG-> pixpath. ‘/ T / delete.gif “alt =” “.
get_string (‘DeletePage’, ‘block_simplehtml’).” /> </ a>’;
Else {}
$ Editar =”;
$ Excluir =”;
}
$ This-> conteúdo> texto .= ‘<li> <a href=”$.CFG-> wwwroot.
‘/ Blocos simplehtml / / view.php? Id =’. $ Simplehtmlpage-> id.
‘& CourseID =’ $ COURSE-> id .”>’.$ simplehtmlpage-> pagetitle ‘</ a>’ $ $ edit delete ‘</
li>’ .....;
}

```

### Criar delete.php

Atualize a página do curso e você deve ver agora um ícone de X preto ao lado de cada página após o ícone de edição (quando a edição é ligado). O objetivo do delete.php é imprimir um aviso para o usuário com uma verificação de que o usuário deseja excluir a página simplehtml. O anúncio irá retornar o usuário para a Claro, se eles dizem não, ou deletar a página e retornar o usuário para a Claro, se eles dizem que sim. Moodle tem uma função de núcleo pré-definidos em weblib.php que irá criar o formulário Sim-Não para você e enviar o usuário para os locais corretos com base na resposta à pergunta. A função é chamada notice\_yesno (). Antes de utilizar este, a configuração da estrutura de arquivos básicos para delete.php (é semelhante ao que tem sido utilizado para a maioria dos outros arquivos como view.php e simplehtml.php).

```

<? Php
require_once (‘../.. / config.php ‘);
CFG $ global, $ USER, $ SITE;

$ CourseID required_param = (‘CourseID, PARAM_INT);
$ Id = optional_param (‘id’, 0, PARAM_INT);
$ Confirm = optional_param (‘confirmar’, 0, PARAM_INT);

if (! $ curso = get_record (‘curso’, ‘id’, $ CourseID)) {
erro (get_string (‘invalidcourse’, ‘block_simplehtml’) $ CourseID.);
}

require_login ($ curso);
require_capability (‘bloco / simplehtml: managepages, get_context_instance
(CONTEXT_COURSE, $ CourseID));

if (! $ simplehtmlpage get_record = (‘block_simplehtml’, ‘id’, $ id)) {
erro (get_string (‘nopage’, ‘block_simplehtml’, $ id));
}

```

```

}
$ Site = get_site ();
print_header (strip_tags ($ site-> fullname), $ local> fullname: "wwwroot href=".$CFG-> <a.
'/ Curso / view.php id=?' $ CourseID ." ">'.$ curso> shortname.. '</ A> ->'.
$ Simplehtmlpage-> pagetitle,"'< meta name = "description" content = ""'.
s (strip_tags ($ local> resumo)) ." ">', verdade,"");
print_footer ();
?>

```

Este código comum:

- \* Verifica o CourseID corresponde a um curso válido
- \* Verifica se o usuário está logado
- \* Verifica se o id página simplehtml corresponde a uma ID de página existente
- \* Imprime o cabeçalho eo rodapé da página

Repare que este é semelhante ao que tem sido utilizado até agora, mas há outro parâmetro opcional “confirmar” foi adicionado.

Esta variável será definido quando o usuário clica no botão Sim ou Não no formulário. Isto significa que um teste pode ser adicionada para testar o valor de retorno, se ele for definido, o registro de dados que contém a ID de página simples deve ser suprimida. Adicione o seguinte código após a todos para o print\_header () função:  
ponto de exclamação

Dica importante!

A maioria das IDEs e editores de texto programador permitem criar modelos para trechos de código comumente usados. Turning seções de código que você usa com frequência em modelos pode economizar muito tempo e digitação!

```

if (! $ confirm) {
$ Optionsno = array ('id' => $ CourseID);
$ Optionsyes = array ('id' = id> $, 'CourseID' => $ CourseID, 'confirmar' => 1, 'sesskey' =>
sesskey ());
print_heading (get_string ('Confirmdelete', 'block_simplehtml'));
notice_yesno (get_string ('DeletePage', 'block_simplehtml', $ simplehtmlpage-> pagetitle),
'delete.php',
$ CFG-> wwwroot '/ curso / view.php', $ optionsyes, $ optionsno, 'post', 'get').;
}
else {
if (confirm_sesskey ()) {
if (! delete_records ('block_simplehtml', 'id', $ id)) {
error ('deleterror', 'block_simplehtml');
}
}
else {
error ('sessionerror', 'block_simplehtml');
}
add_to_log ($ id, 'block_simplehtml', 'Página delete', $ CFG-> wwwroot. "curso / view.php? & id
= '
$ CourseID," $ id, $ USER-> id);
redirect (" $ CFG-> wwwroot / curso / view.php id = $ CourseID?");
}

```

No caso declaração acima, as duas variáveis optionsno e optionsyes estão definidos. Estes são

utilizados para passar as variáveis do formulário para obter ou posto quando o usuário seleciona “sim” ou “não”. Eles são passados para a função `notice_yesno()` como os parâmetros de quinto e sexto. Se o usuário clicar em “não” eles são enviados para a página do curso, o que significa que as necessidades CourseID a ser adicionado ao formulário. Se o usuário clica em “yes” o registro do banco de dados contendo o ID de página simplehtml é excluído eo usuário é encaminhado para o curso. O ID de página, CourseID, confirmar, a chave de sessão e precisam ser passadas para a forma. A chave de sessão é passado para evitar a possibilidade de um cross-site scripting anexar ou usuários que enviam dados de forma válida sem o uso de Moodle.

Agora, dê uma olhada nos parâmetros passados a `notice_yesno()` função. O primeiro parâmetro é o texto que é exibido antes do Sim e Não botões. O segundo parâmetro é a url onde o formulário envia o usuário quando ele clicar o parâmetro sim button. The terceira é a url para onde o usuário será enviado se clicar no botão não. Os parâmetros sexto e sétimo são os métodos HTTP (GET ou Post) usado para enviar os dados do formulário para o sim e não botões.

Limpeza de final de

O bloco está quase pronto! A única coisa que não tem sido feito é a limpeza de todos os dados associados com o bloco de exclusão. Isto é conseguido pela adição de uma função chamada `instance_delete()` para o bloco. Nesta função chamar a excluir registros e usar o blockid passar no \$ this-> instância> variável id. A função fica assim:

```
instance_delete function () {  
delete_records ('simplehtml', 'blockid', $ this-> exemplo-> id);  
}
```

## Unidade 7 – Parte E – definir um papel para o Bloco

### Uma parte – Bloco Básico

1. Criar um simples bloco seguinte tutorial do Moodle. O bloco será nomeado “simplehtml”.
2. Criar um arquivo de linguagem e de diretório.
3. Criar uma tabela de banco de dados em que nosso bloco irá inserir informações.

### Parte B – Formulários

1. Adicionar um formulário usando o Moodle biblioteca de formulários para inserir as informações na tabela
  - \* Definir uma nova forma.
  - \* Apresentar o novo formulário.
  - \* Adicione um cabeçalho.
  - \* Adicione o controle de acesso.
  - \* Adicione registro.
  - \* Adicione elementos de formulário.
  - \* Adicione as variáveis de estado.
  - \* Finalize o formulário.
2. Processar o formulário.

### Parte C – Refinar a Interface de Usuário

1. Adicionar links para cada página do bloco.
2. Adicionar métodos para visualizar páginas.
3. Refactor método de exibição.
  - \* Adicionar título da página.
  - \* Mostrar a data
4. Exibição do texto.
5. Mostrar link para o arquivo.
6. Exibir a imagem.

## Parte D – Adicionar recursos de edição

Adicionar recursos de edição que permitem aos usuários editar dados, incluindo

1. Excluir uma página
2. Adicionar um link para o arquivo delete.php
3. Criar o arquivo delete.php
4. Limpar todos os dados associados com o bloco com o instance\_delete () função

## Parte E – definir um papel para o Bloco

1. Adicionar verificação para o bloco para a capacidade
2. Adicionar um papel para a forma simplehtml
3. Adicionar uma função para editar
4. Adicionar uma função para apagar

Noções básicas de bloqueio | Formulários | Refine the User Interface | Adicionar recursos de edição | definir um papel

Hiperlinks aparecem em novas janelas do navegador.

Com a tabela estabelecida, criar uma capacidade de permitir que o Administrador Moodle para determinar quais funções pode adicionar links para o bloco de html simples. Leia a seção Programming Interface do Moodle docs para ter uma idéia básica de como criar uma capacidade. Criar duas capacidades para este bloco:

1. Ver página simplehtml (s) que serão bloco / simplehtml: viewpages
2. Modificar / adicionar simplehtml página (s) que serão bloco / simplehtml: managepages

Seguindo as instruções na seção Interface de Programação de Moodle docs, crie um arquivo access.php nos blocos / simplehtml diretório / db. Nesse arquivo adicione o array capacidade simplehtml:

```
<? Php
/*
 * Recursos para o bloco simplehtml
 * /
Block_simplehtml_capabilities $ array = (
);
?>
```

Em seguida adicione as duas matrizes capacidades, bloco / simplehtml: viewpages e bloquear / simplehtml: managepages à matriz acima. Cada matriz deve ter os seguintes atributos:

```
* Bloco / simplehtml: viewpages
o tipo: ler
o contexto: curso
o Legacy:
+ Dos visitantes: prevenir
+ Estudante: permitir
+ Professor: permitir
+ Professor Edição: permitir
+ Coursecreator: permitir
+ Admin: permitir
* Bloco / simplehtml: managepages
o tipo: escrever
o contexto: curso
o Legacy:
+ Dos visitantes: prevenir
```

- + Estudante: prevenir
- + Professor: prevenir
- + Professor Edição: permitir
- + Coursecreator: permitir
- + Admin: permitir

No final, o seu access.php deve ser parecido:

```
<? Php
/*
 * Recursos para o bloco simplehtml
 */
Block_simplehtml_capabilities $ array = (
'Block / simplehtml: viewpages' => array (
'Captype' => ler ',
'Contextlevel' = CONTEXT_COURSE>,
'Legado' => array (
'Guest' => CAP_PREVENT,
'Estudante' = CAP_ALLOW>,
'Professor' = CAP_ALLOW>,
'Editingteacher' => CAP_ALLOW,
'Coursecreator' = CAP_ALLOW>,
'Admin' => CAP_ALLOW
)
),
'Block / simplehtml: managepages' => array (
'Captype' => escrever ',
'Contextlevel' = CONTEXT_COURSE>,
'Legado' => array (
'Guest' => CAP_PREVENT,
'Estudante' => CAP_PREVENT,
'Professor' => CAP_PREVENT,
'Editingteacher' => CAP_ALLOW,
'Coursecreator' = CAP_ALLOW>,
'Admin' => CAP_ALLOW
)
)
);
?>
```

Agora bata o número da versão em blocos / simplehtml / block\_simplehtml.php. Mudar o passado, para 0 a 1, onde você definir \$ this-> versão. A linha deve parecer:

```
$ This-> version = 2008010101;
```

Em seguida vá ao link notificações no bloco de Administração do Site. Isso deve atualizar o bloco simplehtml. Verifique se as capacidades foram adicionadas, indo para Usuários> Permissões> Definir papéis e selecionando Administrador. Pesquisar simplehtml. papéis definidos estão incompletos

O que faz o simplehtml: viewpages significa? A presença de uma cadeia cercada por colchetes duplos quadrados implica que há uma string referenciada no código que ainda não foi definido em um arquivo de idioma apropriado. Adicione o seguinte ao lang/en\_utf8/block\_simplehtml.php:

```
$ String ['simplehtml: viewpages'] = 'Ver páginas HTML simples';
```

```
$ String ['simplehtml: managepages'] = 'Gerenciar páginas HTML simples';
```

Com a fixa problema, o Definir página Funções irá agora olhar como:

papéis definidos estão agora completa com a sequência de texto adicionado ao arquivo de idioma Nesta parte da unidade, até agora, você aprendeu como criar um bloco básico com todas as funções padrão que Moodle precisa exibir o bloco com alguma configuração limitada. Você aprendeu como criar uma tabela de banco de dados usando o editor de XMLDb no Moodle, acrescentou um novo ficheiro de linguagem para o bloco, e criou um bloco de capacidades que podem ser usados para determinar se um usuário tem as permissões corretas. Em seguida, você vai trabalhar na criação de um formulário para adicionar informações à tabela do banco de dados que você criou.

Adicionar verificação para o bloco para a capacidade

Com os dois novos recursos criados, é hora de usá-los e adicionar um link para o formulário de soon-to-be-criado no bloco. O lugar mais fácil de adicionar o link está no rodapé, mas é fundamental que a pessoa que vê este link tem a capacidade de gerenciar página simplehtml. Modificar o get\_content () em blocos / simplehtml / block\_simplehtml.php para verificar o cheque capabilities.To correto para os recursos, use o has\_capability () função encontrada em lib / access.php. Mas, primeiro adicione o mundial CURSO variável \$ e \$ CFG para a função usando:

```
CURSO global $, $ CFG;
```

Para verificar se o usuário tem as permissões visão correta substituir:

```
$ This-> content-> text = $ this-> config-> texto;
```

com

```
$ Context = get_context_instance (CONTEXT_COURSE, $ COURSE-> id);
```

```
if (has_capability ('bloco / simplehtml: viewpages', $ contexto)) {
```

```
$ This-> content-> text = $ this-> config-> texto;
```

```
Else {}
```

```
$ This-> content-> text ="";
```

```
}
```

Para adicionar o link para o formulário substituir:

```
$ This-> conteúdo> rodapé ="";
```

com:

```
if (has_capability ('bloco / simplehtml: managepages', $ contexto)) {
```

```
$ This-> conteúdo> rodapé = '<a href=""'. $CFG-> wwwroot.
```

```
'/ Blocos / simplehtml / simplehtml.php? Blockid ='.
```

```
$ This-> exemplo-> id. '& CourseID =' . $ COURSE-> id . ''>'.
```

```
get_string ('addPage', 'block_simplehtml').</ a>';
```

```
Else {}
```

```
$ This-> conteúdo> rodapé ="";
```

```
}
```

Seu get\_content final () deve ser parecido:

```
get_content function () {
```

```
CURSO global $, $ CFG;
```

```
if ($ this-> conteúdo! == NULL) {
```

```
return $ this-> conteúdo;
```

```
}
```

```
$ This-> content = new stdClass;
```

```
$ Context = get_context_instance (CONTEXT_COURSE, $ COURSE-> id);
```

```

if (has_capability ('bloco / simplehtml: viewpages', $ contexto)) {
$ This-> content-> text = $ this-> config-> texto;
}
else {
$ This-> content-> text = "";
}

if (has_capability ('bloco / simplehtml: managepages', $ contexto)) {
$ This-> conteúdo = footer "<a href=" . $CFG-> wwwroot. ' / Blocos / simplehtml / page.php?
Blockid ='.
$ This-> exemplo-> id. '& CourseID =' . $ COURSE-> id . "'>".
get_string ('addPage', 'block_simplehtml').'/>';
}
else {
$ This-> conteúdo = rodapé = "";
}

return $ this-> conteúdo;
}

```

Observe, enquanto a construção da ligação que \$ this-> exemplo-> id é usado. Este é um identificador que é único para cada instância do bloco em um curso e permite a utilização de vários blocos em um curso. CourseID também é usado, junto com o ID de instância do bloco, para mais facilmente encontrar todas as páginas associadas ao bloco para um curso. O link para page.php também é enviado, que é inexistente, que serão adicionados com o código necessário na próxima seção.

Enquanto a construção de nosso link, get\_string () também é usado, então uma entrada para 'addPage' a corda deverá ser adicionada ao arquivo de idioma para o bloco com texto de sua escolha.

Finalmente, atualize o curso Moodle para o qual você adicionou o bloco simplehtml. Como um administrador ou professor, você deve ver o "addPage 'link; como estudante você não deve ver o link.

Adicionar ao formulário de papel simplehtml

Atualmente qualquer usuário logado pode usar o formulário simplehtml. No desenvolvimento isso não é tão grande de uma preocupação, mas antes de passar a produção de introdução sempre assegurar que os usuários tenham acesso apenas às partes do aplicativo que deveriam. Em page.php após require\_login (\$ curso); adicione a seguinte linha:

```

require_capability ('bloco / simplehtml: managepages, get_context_instance
(CONTEXT_COURSE, $ CourseID));

```

Isso garante que o usuário não pode acessar esta página, sem ter 'block / simplehtml: managepages "a capacidade.

Add Role para Edição

Restringir a edição para usuários que tenham o bloco / simplehtml: capacidade managepages adicionando controle de acesso à ação de edição. Em edit.php usar a variável \$ canmanage para determinar o acesso. Definir esta variável, verificando o usuário tem a capacidade adequada.

```

$ Canmanage isediting = ($ curso-> id) & &
has_capability ('bloco / simplehtml: managepages, get_context_instance (CONTEXT_COURSE, $
CourseID));

```

Adicionar Função para Excluir

Para restringir exclusão para aqueles que têm a capacidade managepages, adicione a seguinte linha



```
após delete.php require_login ($ curso);  
require_capability ('bloco / simplehtml: managepages, get_context_instance  
(CONTEXT_COURSE, $ CourseID));
```

## Apêndice A: Convenções de nomenclatura Moodle

Recursos externos abertos em novas janelas do navegador.

Moodle tem algumas convenções de nomenclatura comumente observados. Embora você está livre para nomear arquivos que quiser, isso não é recomendado. Algumas diretrizes para nomear arquivos são:

- \* Nome do arquivo depois de a função que está realizando, como atualizar, editar, apagar ou criar
- \* Nome arquivos de idioma para os blocos de partida com block\_ para evitar colisões ao usar o get\_string () e print\_string () funções
- \* Nome arquivos PHP que criar uma classe de formulário Moodle com o sufixo de form.php para indicar que elas são uma forma. Exemplo: simplehtmlform.php
- \* Os arquivos PHP que faz processo deve ser nomeado após o processo de forma que, sempre que possível, ou a ação que eles tomam. Exemplo: simplehtml.php processa a simplehtml\_form edit.php e edita o simplehtml\_form
- Biblioteca de arquivos \* deve ser nomeado ou lib.php locallib.php

Documentação do Moodle desenvolvedor tem mais informações sobre convenções para trabalhar com o Moodle.

- \* Guia de Estilo Código
- \* Guia de Interface do Usuário
- \* Teste de unidade

## Unidade 7 – Apêndice B – O Editor XMLDb

Moodle oferece uma interface gráfica de usuário (GUI) para criar o arquivo XML. Para referência, o arquivo criado pelo XMLDb Moodle do GUI é chamada install.xml e é armazenado no diretório db de um bloco ou diretório do módulo base (blocos de exemplo / simplehtml / db). Você pode olhar para o GUI, entrando em sua instalação Moodle e clicando no link “Diversos” no bloco de Administração do Site. Em seguida, clique em XMLDb.

vista principal da X.M.L.D.B. editor  
ponto de exclamação

Dica importante!

Se a Criar ou textos de carga não são azuis (indicando um link ativo) ea pasta db e / ou o arquivo install.xml existe isso significa que o usuário que iniciou o Apache não tem permissão para ler o diretório. Isso não deve ser um problema em um computador com Microsoft Windows, mas poderia ser em um computador com Mac OS X ou Linux.

Haverá uma linha para cada tipo de módulo de bloco, e pergunta instalado em sua instância Moodle, que tem um diretório db. Aqueles com o Criar em azul têm uma ligação ativa e conter um diretório db no diretório base blocos. Os links que têm o seu nome (blocos de exemplo / admin / db) eo texto Load em azul (link ativo com o tema padrão) tem um arquivo no diretório install.xml db listados.

Explore o XMLDb agora. Clique ao lado de blocos de carga / rss\_client / db. A página irá recarregar e os blocos / rss\_client / db linha é realçada em verde. Editar e Unload são links ativos agora e de carga e Excluir são inativos. Em seguida clique no link Editar.

editar Ver arquivo xml de X.M.L.D.B. editor

Aqui você vê um campo para os comentários para o arquivo XML e todas as tabelas relacionadas com o arquivo do RSS install.xml bloco cliente. (Existe apenas um block\_rss\_client.) Você também vai ver os links para mudar a sua visão para a tabela original se você tiver editado o XML Ver Original. Para mudar a sua visão para o XML enquanto editado no modo de exibição original, clique em Exibir Editado, que fornece as seguintes opções:

- \* Criar uma tabela nova tabela New
- \* Criar uma tabela de mysql Tabela novo código de MySQL
- \* Declaração de criar nova declaração Nova
- \* Voltar para a última página Voltar à página principal
- \* Ver o código SQL para criar todas as tabelas listadas Ver SQL Código
- \* Ver o código php para criar todas as tabelas listadas

ponto de exclamação

Dica importante!

Não provedor de cada banco de dados usa os tipos mesmo campo, nem são todos os tipos de dados presentes em todos os bancos de dados. Por exemplo, o SQL Server usa um tipo de dados bit para valores booleanos e Oracle não implementa um tipo boolean ou enum em tudo. A GUI XMLDb entende isso e resumos dos tipos de dados para que o código do Moodle pode lidar com os detalhes da criação da tabela para bancos de dados individual. Desenvolvedor pode então usar a API de banco de dados, que foi coberto na Unidade 5, para acessar as informações da tabela sem saber exatamente como cada campo é criado em cada banco de dados.

\* Clique no link Editar ao lado block\_rss\_client para ver a definição de tabela para block\_rss\_client. Agora você deve ver um campo com o nome do bloco, e um para os comentários associados com a tabela.

\* O [Ver Original] link muda a sua visão para a tabela original se você tiver feito alterações no XML.

\* O [Ver Editado] link muda a sua visão para o XML editado se você estiver no modo de exibição original.

\* O [novo campo] link irá adicionar um novo campo para o final da lista de campos.

\* A tecla [New] link irá adicionar uma nova chave primária, única ou estrangeiras.

\* O [Novo Índice] link irá adicionar um novo índice.

\* O [Voltar] link navegar de volta para a lista de tabelas no arquivo install.xml.

\* O [Ver Código SQL] link exibe o código SQL para criar a tabela para o MySQL, Oracle e PostgreSQL.

\* O [Ver Código PHP] link exibe o código PHP para adicionar, excluir e renomear, bem como campos de mudança, chaves ou índices.

Abaixo desses links, você verá uma listagem de todos os campos na tabela block\_rss\_client juntamente com links para editar Editar, mover o campo de cima ou para baixo para baixo na definição da tabela, e excluir o campo Delete. A última coluna na página é a definição do campo.

Você deve perceber que a definição um tanto utiliza campos padrão SQL como texto, int, char.

Abaixo a definição campos existe uma área para Chaves, que se refere ao primário, chaves únicas e estrangeiros. Finalmente abaixo do Chaves há uma área onde os índices para a tabela existe. Para a tabela block\_rss\_client não há índices.

Clique em Editar ao lado de userid. Você verá um formulário HTML para configurar o campo na tabela. Os campos devem ser auto-explicativo:

- \* Nome do campo
- \* Comentários sobre o campo
- \* Digite a

- \* Comprimento (cada tipo tem diferentes tamanhos, que será exibido ao lado do campo de comprimento quando o tipo é alterado)
- \* Número de casas decimais para um tipo de número
- \* Assinado ou não assinado para um tipo inteiro
- \* Nulo ou não nulo
- \* Sequência para um tipo inteiro
- \* Enum ligado ou desligado por um personagem ou tipo de texto
- \* E um valor padrão

A seguir são links para ver os valores originais para o campo, se tiver alterado os Ver Original, ver a versão editada do campo quando no modo de exibição original editado, e para navegar de volta para a definição da tabela Back. Agora navegar de volta para fora do editor XMLDb clicando Back, Back, Back To Main. Se nós modificamos os blocos / rss\_client / db arquivo install.xml antes voltamos para a página principal XMLDb os links Salvar e Reverter estariam ativos além Editar. Se você quiser manter as alterações que você fez, você deverá clicar em Salvar o arquivo e install.xml teria de ser alterado. Se você não deseja manter as alterações que você deverá clicar em Reverter eo install.xml seria recarregado a partir install.xml. Finalmente descarregar o arquivo install.xml clicando no Unload.

#### Apêndice D: A atualização de um Bloco

As três primeiras partes desta Unidade de ter coberto como desenvolver um simples bloco desde o início. No entanto, que ações devem ser tomadas a fim de atualizar um bloco se você quiser adicionar um campo de banco de dados ou tabela ou um novo recurso? Esta parte fornece detalhes sobre como fazer essas atualizações, incluindo:

1. Como adicionar um índice novo banco de dados em uma tabela
2. Os passos para instalar o campo para um bloco pré-existent

Esta Unidade tem como objetivo fornecer mais do que você precisa saber sobre o desenvolvimento de um bloco Moodle. Depois de concluir todas as atividades e atribuições para esta unidade, você terá uma base sólida sobre como iniciar um bloco a partir do zero, adicionar uma tabela de banco de dados para o Moodle, criar um formulário Moodle, e atualizar um bloco com suas alterações.

“To Do” Lista

1.

Pratique editando a tabela de banco de dados.

edit ecrã de índice shotAdding um novo índice ou campo para a tabela do banco de dados é feita usando o editor XMLDb. Navegue até o editor XMLDb e carregar o arquivo install.xml para blocos / simplehtml / db. Em seguida clique no link “Editar” e depois clique no link “Editar” ao lado simplehtml. Clique no link “Novo índice”. Para criar um índice em blockid adicione as seguintes informações para cada um dos campos:

- \* Nome: index\_blockid
- \* Comentário: índice para blockid
- \* Tipo: não exclusivo
- \* Campos: blockid

Clique no botão “Alterar”. Deve haver um índice na parte inferior da página Editar Tabela chamado index\_blockid.

Antes de salvar o upgrade.php install.xml precisa ser criado. Quando um arquivo install.xml existe na pasta db para um bloco eo bloco foi instalado já então Moodle irá chamar upgrade.php quando o usuário clica no link Notificação no bloco de Administração do Site, se a versão para o bloco foi atualizado . Isso significa que o desenvolvedor precisa para adicionar o upgrade.php para a pasta db e aumentar o número de versão do bloco para o novo índice a ser instalado em um site Moodle que

tenha previamente instalado o bloco simplehtml. Adicionar upgrade.php para a pasta db agora e cole o seguinte código no arquivo:

```
<? Php
// Este arquivo mantém o controle de atualizações para
// O bloco de pesquisa
//
// Às vezes, as mudanças entre as versões envolvem
// Alterações às estruturas de banco de dados e outros
// Coisas importantes que podem quebrar instalações.
//
// A função de atualização neste arquivo tentará
// Para executar todas as ações necessárias para atualização
// Sua instalação mais antiga para a versão atual.
//
// Se há algo que não pode fazer em si,
// Vai dizer o que você precisa fazer.
//
// Os comandos serão todos aqui do banco de dados neutro,
// Usando as funções definidas no lib / ddllib.php
xmldb_block_simplehtml_upgrade função ($ oldversion = 0) {
CFG global $, $ THEME, $ db;
$ Result = true;
/// E de upgrade começa aqui. Para cada um, você precisará de um
// Bloco / de código semelhante ao seguinte. Por favor, apague
/// Este linhas de comentário uma vez que este arquivo iniciar tratamento adequado
/// Atualização de código.
return $ resultado;
}
?>
```

Clique em Ver Código PHP no editor XMLDb. Na gota de ação seleção para baixo selecionar índice adicionar. Em “Select Field / Key / Index” “index\_blockid.” Select Clique no botão “Exibir”. Copie o código na área de texto HTML e colá-lo em upgrade.php após a linha contendo “código // upgrade /”. Agora mudar “XXXXXXXXX” para a data atual no formato YYYYMMDD00. Este é o número de versão que utiliza o Moodle para determinar se algo precisa ser atualizado. Por exemplo, o número da versão 2008010100 representa 01 de janeiro de 2008. Copie esse número e cole no block\_simplehtml.php onde você encontra:

```
$ This-> version = 2004111200;
```

Deve ser algo como:

```
$ This-> version = 2008021900;
```

O seu código completo deve ser algo como:

```
<? Php
// Este arquivo mantém o controle de atualizações para
// O bloco de pesquisa
//
// Às vezes, as mudanças entre as versões envolvem
// Alterações às estruturas de banco de dados e outros
// Coisas importantes que podem quebrar instalações.
//
// A função de atualização neste arquivo tentará
```

```
// Para executar todas as ações necessárias para atualização
// Sua instalação mais antiga para a versão atual.
//
// Se há algo que não pode fazer em si,
// Vai dizer o que você precisa fazer.
//
// Os comandos serão todos aqui do banco de dados neutro,
// Usando as funções definidas no lib / ddllib.php
xmldb_block_simplehtml_upgrade função ($ oldversion = 0) {
CFG global $, $ THEME, $ db;
$ Result = true;
/// E de upgrade começa aqui. Para cada um, você precisará de um
// Bloco / de código semelhante ao seguinte. Por favor, apague
/// Este linhas de comentário uma vez que este arquivo iniciar tratamento adequado
/// Atualização de código.
if ($ result & & $ oldversion <2008021900) {
/// Define índice index_blockid (não único) para ser adicionado à block_simplehtml
Tabela $ = new XMLDBTable ('block_simplehtml');
$ Index = new XMLDBIndex ('index_blockid');
$ Index-> setAttributes (XMLDB_INDEX_NOTUNIQUE, array ('blockid'));
/// Lançamento adicionar índice index_blockid
$ Result = result & & $ add_index ($ table, $ index);
}
return $ resultado;
}
?>
```

Índices shotClick ecrã da lista Voltar> Voltar> Voltar para a principal> Salvar para blocos / simplehtml / db. Por fim, clique em “Notificações” para ver o bloco atualizado. Verifique se o bloco foi atualizado por olhar para o seu banco de dados phpmyadmin. Localize a tabela “mdl\_block\_simplehtml”. Clique no botão “Estrutura” tab no phpmyadmin. Na seção de índices para o fundo da página que você deve ver agora a “mdl\_simp\_blo\_ix índice.”

A partir de agora, ao adicionar uma cópia do novo campo “if” de cima e mudar a versão para a versão atual. Insira o código PHP a partir do editor XMLDb para o campo ou informações que estão sendo adicionados dentro do “if”.

2.

Completar a tarefa de adicionar um campo novo banco de dados para a tabela simplehtml para manter o nome de um animal que o usuário escolhe em uma caixa de Select.

Certifique-se de usar o Editor de XMLDb para esta tarefa.

1. Adicionar um campo que vai conter a localização do animal. O local terá um limite de 100 caracteres.

2. Adicionar um novo grupo para a forma de conter o novo campo.

3. Adicionar uma gota Selecione para baixo e rótulo ao formulário simplehtml para o nome do animal. Adicionar os seguintes nomes para a lista de seleção:

- \* Tiger
- \* Lion
- \* Urso
- \* Donkey
- \* Macaco

4. Adicionar um campo de texto com o rótulo ao formulário simplehtml para a localização do animal é hospedado.

5. Mostrar a dois novos campos na página simplehtml abaixo do visor segundo da imagem.

6. Modificar db / upgrade.php para adicionar os novos campos para o banco de dados.
7. Ajustar o número de versão para o bloco.
8. Apresentar uma cópia compactada do seu bloco simplehtml para a área de atribuição designada.

## Unidade 8: Teste de Métodos

### Visão geral | Unidade 8 “fazer” lista

Recursos externos abrem em novas janelas do navegador.

Software Testing e Q.A. (Quality Assurance) é um grande tema, embora também seja um tema muito importante para todos os programadores. As grandes empresas costumam ter um software dedicado QA engenheiros responsáveis apenas para teste de software ou QA.

Apesar de crucial para a qualidade de software e amplamente utilizados por programadores e testadores, teste de software ainda é uma arte, devido à limitada compreensão dos princípios de software. A dificuldade em testes de software decorre da complexidade do software: não podemos testar um programa completamente com complexidade moderada. [Engenharia de Software. (2002). Hagen:.. Fernuniv, Fachbereich Informatik]

O teste é mais do que apenas a depuração. O objetivo do teste pode ser garantia de qualidade, verificação e validação, ou estimativa de confiabilidade. O teste pode ser usado como uma métrica genérica também. Testes de correção e testes de confiabilidade são duas grandes áreas de teste. Teste de software é um trade-off entre tempo, orçamento e qualidade. Teste de Software [. (1999). Pan: Carnegie Mellon University] Mais detalhes sobre a garantia de qualidade de software..

Nesta unidade, vamos nos concentrar em alguns tópicos que são particularmente importantes para os desenvolvedores Moodle.

#### Testes de caixa preta

Segundo a Wikipedia (2-14-08):

“Testes de caixa preta toma uma perspectiva externa do objeto de teste para derivar casos de teste. Esses testes podem ser funcionais ou não funcionais, embora geralmente funcional. O designer de teste seleciona entrada válidos e inválidos e determina a saída correta. Não há conhecimento de estrutura interna do objeto de teste.

Este método de design de teste é aplicável a todos os níveis de teste de software: unidade, integração, teste funcional do sistema, e aceitação. Quanto maior o nível, e, portanto, o maior e mais complexa a caixa, mais um é forçado a usar testes de caixa preta para simplificar. Enquanto este método pode revelar partes unimplemented da especificação, não se pode ter certeza de que todos os caminhos existentes são testados. “

Usuários podem querer avaliar ou comparar as novas versões do Moodle antes de tomar uma decisão final sobre a atualização. O “teste de caixa preta” do novo código base do ponto de vista de um usuário final – identificando possíveis problemas – pode ser valioso.

#### Teste de caixa branca

Segundo a Wikipedia (2-14-08):

“O teste de caixa branca (teste de caixa aka claro, testes de caixa de vidro ou de teste estrutural) usa uma perspectiva interna do sistema para projetar casos de teste com base na estrutura interna. Exige habilidades de programação para identificar todos os caminhos através do software. O testador escolhe caso de teste insumos para o exercício caminhos percorrer o código ea determinar a outputs apropriados ... Como os testes se baseiam na aplicação real, se as mudanças, execução dos testes, provavelmente também precisará. “

Por exemplo, se uma função importante no Moodle questionário mudanças módulo, então a maioria dos casos de teste do quiz, provavelmente, terão de ser revistas também.

“Embora o teste de caixa branca é aplicável na unidade, a integração e os níveis do sistema do processo de teste de software, é normalmente aplicada para a unidade. Embora normalmente os testes caminhos dentro de uma unidade, ele também pode testar os caminhos entre as unidades durante a integração, e entre subsistemas durante um teste de nível de sistema. Embora este método de design de teste pode revelar um número esmagador de casos de teste, pode não detectar partes unimplemented da especificação ou requisitos em falta, mas pode ter certeza de que todos os caminhos através do objeto de teste são executados. “

Teste de regressão

Segundo a Wikipedia (2-14-08):

“O teste de regressão é qualquer tipo de teste de software, que procura descobrir erros de regressão. Bugs de regressão ocorrer sempre que a funcionalidade de software que trabalhou anteriormente como desejado, pára de funcionar ou já não funciona da mesma forma que anteriormente estava programado. Normalmente os erros de regressão ocorrem como uma involuntária consequência de alterações no programa.

Métodos comuns de testes de regressão incluem re-execução de testes executados anteriormente e verificar se as falhas anteriormente fixado reapareceram. “

Um exemplo típico do uso de testes de regressão é verificar se corrige bug novo no Moodle realmente introduziu outros novos bugs. Outros do que a forma ‘manual’ de testes, você pode encontrar usando ferramentas de teste automatizado, tais como “Selenium IDE”, com alguns casos pré-estabelecidos teste seria muito útil para verificar se os casos de teste ainda se comportam da mesma maneira. “Selenium IDE” será explorada em mais detalhes posteriormente.

topo

Teste de unidade

Segundo a Wikipedia (2-14-08):

“Na programação de computadores, testes de unidade é um procedimento usado para validar unidades individuais de código-fonte estão funcionando corretamente. A unidade é a menor parte testável de um aplicativo. Na programação procedural uma unidade pode ser um programa individual, função, procedimento, etc , enquanto em programação orientada a objeto, a menor unidade é um método, o que pode pertencer a uma base / super classe, classe abstrata ou classe derivada / criança “.

Uma vez que o teste de unidade deu-lhe a confiança que cada parte funciona, então você deve utilizar outras formas de testes para garantir que as diferentes partes trabalham juntas de forma adequada.

Moodle foi construído em um quadro de unidade de teste que se baseia em um quadro SimpleTest. Mais tarde vamos discutir como utilizar este framework.

Referem-se a esses recursos úteis para obter informações adicionais: “o teste de unidade” em geral, técnicas de teste de design, testes Moodle da unidade.

Próprio do Moodle “Os testes de unidade” é uma ferramenta para desenvolvedores Moodle para avaliar códigos Moodle e é encontrada no bloco de Administração em Relatórios quando logado como administrador do sistema Moodle.

criação de um teste de unidade no Moodle

Testes de unidade pode ser um dos primeiros passos em um processo de controle de qualidade para o desenvolvimento ou tweaking o código do Moodle.

Casos de selênio

De acordo com OpenQA (2-14-08):

“Selenium IDE é um ambiente de desenvolvimento integrado para os testes com Selenium. Ele é implementado como uma extensão do Firefox e permite gravar, editar e depurar os testes. Selenium IDE inclui o Selenium núcleo inteiro, permitindo que você facilmente e rapidamente gravar e reproduzir testes em ambiente real que será executado.

Selenium IDE não é apenas uma ferramenta de gravação: é uma IDE completa. Você pode optar por usar a sua capacidade de gravação, ou você pode editar seus scripts com a mão. Com o apoio autocomplete ea capacidade de se mover em torno de comandos rapidamente, Selenium IDE é o ambiente ideal para criar os testes com Selenium não importa o estilo de testes que você preferir.

Características do Selenium IDE são

- \* Fácil registro e reprodução
- \* Seleção de campo inteligente irá utilizar identificações, nomes ou XPath, conforme necessário
- \* Autocomplete para todos os comandos comuns de selênio
- \* Ande através de testes
- \* Debug e definir pontos de interrupção
- \* Salvar como HTML testes, scripts Ruby, ou qualquer outro formato
- \* Suporte para Selenium arquivo user-extensions.js
- \* Opção para automaticamente fazer valer o título de cada página “

Para usar o Selenium IDE para testar Moodle, você pode gravar uma seqüência de interações com o seu site Moodle e, posteriormente, reproduzi-los e ver se eles continuam a trabalhar depois de ajustar alguns dos códigos do Moodle.

Existem algumas limitações para a aplicação Selenium IDE Moodle:

- \* O selênio não vai esperar por ‘Continuar’ redireciona para acontecer. Ao gravar uma seqüência de ações, você precisa clicar nos links continuar manualmente.
- \* Por padrão, ele tende a registrar ações usando ids, e em Moodle estes, por vezes, contêm ids banco de dados arbitrários. Por exemplo, envio de uma pergunta do quiz, ele irá gravar `clickAndWait resp94_submit`. No entanto, você pode editar isto para dizer `clickAndWait // input[@ valor = 'Enviar']`.
- \* Em outras ocasiões, por padrão, não é específico o suficiente. Por exemplo, excluir um curso, ele grava `// a [img / @ alt = 'Delete']`, que só vai excluir o primeiro curso listados! Novamente, você pode editá-lo para algo como `// a [img / @ alt = 'Delete' e .. / .. / td [1] / a = 'selênio curso de teste do quiz']`, que irá apagar um curso chamado.
- \* Às vezes não funciona com janelas pop-up mesmo que ele deve apoiá-los.

Informações adicionais sobre como usar Selenium para projetar alguns testes automatizados para Moodle, está disponível no Moodle “Quiz e perguntas dia de testes da comunidade” e no pós fórum de discussão por Tim Hunt, “selênio [sic] para testes Moodle”.

topo

“To Do” Lista

Como parte das atividades desta unidade, siga as instruções abaixo para (1) executar o básico “unidade de teste” testes, (2) escrever o seu código de teste própria unidade, e (3) criar um caso de selênio.

1.

Executar testes de unidade básica.

Use estas etapas para executar uma base Moodle “unidade” de teste.

1. Entrar com uma conta de administrador.
2. Ir para o bloco de Administração do Site
3. Clique no link Relatórios perto da parte inferior do bloco, no canto inferior direito.
4. Clique no link Os testes de unidade
5. Clique no botão Executar testes. (Você pode selecionar as opções que você quer aqui, vamos



explicar estas opções abaixo).

6. Aguarde os testes para ser concluído.

Opções para Executando os testes

Mostrar \* passes, assim como falha. Normalmente, apenas detalhes dos testes que falharam são impressas. Ativando esta opção mostra detalhes de todos os passes bem sucedidos também.

\* Mostrar a busca de arquivos de teste. Os testes a serem executados são encontrados automaticamente busca na base de código para os arquivos cujos nomes correspondem teste \*. php em diretórios chamados simpletest. Ativando esta opção irá imprimir uma lista das pastas pesquisadas e os arquivos de teste encontrado. Isso às vezes é útil para depuração, mas é particularmente útil quando um de seus arquivos de teste tem um erro de sintaxe. Quando isso acontece, você às vezes, é só pegar uma página em branco sem nenhuma mensagem de erro. Ligar a opção de pesquisa mostram que permite que você veja arquivo de teste que foi que deu o erro. Se necessário, você pode ativar essa opção manualmente, acrescentando que “showsearch = 1” ao final da URL.

\* Executar um teste completo (pode ser lento). Se você ativar esta opção, então, bem como à procura de arquivos chamados teste \*. php, a pesquisa também procura por arquivos chamados slowtest \*. php. Para ser útil, o teste completo deve encontrar mais bugs, mas não leva muito tempo para ser concluído. Então se você tem testes muito, muito detalhada de uma área do código, pode ser melhor para selecionar um subconjunto de testes everyday, e só usar os testes mais detalhados quando um erro é informado, ou você está fazendo novo desenvolvimento nessa área do código. Esta opção é mais útil quando combinada com a próxima opção.

\* Somente executar testes em [textfield]. Normalmente, os testes de todas as partes do codebase são executados. No entanto, quando você está apenas fazendo o desenvolvimento de uma parte do código, isso é um desperdício de tempo. Você pode digitar o nome de um diretório (por exemplo mod / quiz) ou um arquivo de teste específico (por exemplo lib / simpletest / testdatalib.php) e, em seguida, apenas os testes serão executados.

```
Pass: lib / simpletest / testdatalib.php -> datalib_test ->
test_get_record -> False False xxxxxxxxxx obter a
[Var/www/html/tjh238/head/lib/simpletest/testdatalib.php / linha
92]
```

Em vez de digitar um caminho nesta caixa, não há uma maneira mais fácil. Sempre que uma aprovação ou reprovação é exibido, o nome do arquivo de teste é impresso. Cada seção do nome do caminho é um link para executar apenas os testes em que pasta ou arquivo.

topo

2.

Escrever código de teste

Continuando com o exemplo “simplehtml” da Unidade 7, escreva um teste de unidade para o block\_simplehtml\_convert\_unixtime função no arquivo / blocos / simplehtml / lib.php.

Para obter uma melhor compreensão de como criar códigos para testes de unidade, procure por esses arquivos de teste simples exemplo:

lib / simpletest / testweblib.php

lib / simpletest / testmoodlelib.php

tipo de pergunta / / SHORTANSWER / simpletest / testquestiontype.php

Crie um diretório chamado / blocos / simplehtml / simpletest e criar um arquivo nele chamado test\_lib.php. O conteúdo real deste arquivo deve ser parecido:

```
<? Php
require_once (dirname (__FILE__) config.php'../..../.);
global $ CFG;
require_once ($ CFG-> libdir 'simpletestlib.php /');
classe basicgrouplib_test estende UnitTestCase {
```

```

test_block_simplehtml_convert_unixtime function () {
// Escreve o código de teste aqui. Isso provavelmente envolve,
// Chamando alguns métodos da classe bloco,
// E então chamar um ou mais $ this-> afirmar ... métodos.
}
}
?>

```

Ou seja, você tem uma classe chamada `something_test`, e nessa classe você tem muitos métodos chamados `test_something`. Normalmente, há um método de teste para cada função para testar, e assim como você pode chamar o método de teste `test_name_of_function_being_tested`. Neste exemplo, este método pode ser chamado `test_block_simplehtml_convert_unixtime`.  
Como escrever documentos caso de teste

Moodle tem um padrão preliminar para construir casos de teste. Estes casos de teste, basicamente, são alguns “cenários” que deve ser executado no Moodle antes de cada lançamento. Atualmente, esse plano de teste está longe de ser perfeita, apesar de mais pessoas na comunidade Moodle estão começando a contribuir mais casos de teste sobre a documentação Moodle wiki (<http://docs.moodle.org>).

Os cenários já existentes deve ser executado no Curso baixado Moodle de teste. Aqui estão algumas regras gerais:

\* Caso de Teste Naming convenções:

o [AAA]: Três personagens que representam o bloco ou o nome do módulo

o [000]: Três dígitos de um número que questionam a

o [A]: Potencial actor = S: professor, A:: estudante, T administrador

\* Os planos de testes são classificados por módulos:

Testes o / Atribuição

Testes o / Chat

Testes o / Escolha

Testes o / Forum

Testes o / Quiz

Testes o / Resource

topo

Usando “simplehtml” como um exemplo, que se segue é um caso de teste típico:

[Sim001TAS]: Adicionando bloco Simplehtml do bloco de menu pull-down

1. Clique no botão “Activar modo edição” botão na janela do canto superior direito.
2. Blocos do pull-down menu, encontrará o bloco “simplehtml” na lista de blocos disponíveis
3. Escolha o bloco “simplehtml” no menu
4. Faz o display block “simplehtml” corretamente? O padrão vai aparecer no lado direito da janela do navegador. Não deve haver cordas faltando, etc

bloco HTML simples exibidas na página principal da disciplina

Mais informações sobre como escrever um documento Moodle caso de teste está disponível no Moodle Docs.

topo

3.

Tente Selenium.

1. Leia OpenQA está usando Selenium IDE e gravação de um teste (note o filme que ilustra as etapas para o teste perto do topo sob a guia “Exibir”).
2. Instale Selenium IDE da Selenium Abra QA da página de downloads IDE.

3. Baixe o arquivo de exemplo Selenium Moodle e descompactá-lo (por exemplo, em c: \xampp \ moodle \)
4. No Firefox, vá para a sua instalação moodle local (por exemplo, Http://localhost/moodle/)
5. Entrar como admin
6. Iniciar Selenium IDE a partir do menu ferramentas do Firefox.
7. A partir do menu Selenium, selecione Arquivo-> Abrir e abra o arquivo “Createquiz.selenium.html” do arquivo descompactado no passo 2.
8. Na lista de comandos, olhe para o 1 / 6 a partir do final – onde diz “tipo newfile c: \ Documents and Settings \ TJH239 \ Desktop \ testes Selenium \ suite.xml questão de teste.” Mudança que o caminho do arquivo para apontar para sua cópia de “suite.xml questão Test” do arquivo descompactado.
9. Clique no botão play verde na barra de ferramentas Selenium. O script de teste será automaticamente:
  - \* Criar um novo curso
  - \* Adicionar um quiz para que
  - Import \* 12 perguntas da amostra
  - \* Adicionar essas perguntas para um questionário
10. Em seguida, ‘Teste de harmonização – fundamental no questionário preview.selinium.html’ open e executá-lo (sem edição necessários neste momento). Este é executado através de uma das perguntas do teste algumas vezes.
11. Finalmente, abra “teste Excluir course.selinium.html”, e executá-lo para remover o curso de teste e voltar para onde você começou.
12. Rever o “suite.xml questão Test” para ver o que é faz, tentar ajustá-lo um pouco para entender como funciona Selenium.

4.

Completar os três Atribuições e submetê-los nas áreas de atribuição designada.

1. [Caso de Teste] Siga o Moodle convenção caso de teste para escrever um caso de teste simples (cenário) para testar Gradebook Plus versão 2 (aka GBPV2).

Considere-se um foco em testes uma das características básicas em GBPV2. Por exemplo, adicionar um evento classificado, usando o “grau de editar” guia com out-of-bound valores, etc Envie seu caso de teste como um documento de texto on-line na área de atribuição. Convenções de nomenclatura e formato para o caso de teste Moodle devem ser observados.

2. [Selenium] Design um caso Selenium automaticamente:

- \* Use a sua instalação Moodle desenvolvimento local (Moodle 1.8).
- \* Entrar como Admin.
- \* Selecione o Curso Moodle caso de teste que você instalou antes.
- \* Clique em “Grades” (livro de notas).
- \* Clique no botão de “Download em formato Excel” botão
- \* Verificar manualmente o conteúdo para ver se tudo funciona corretamente.
- \* Escreva um trecho de código simples usando o Moodle “testes de unidade” quadro para testar uma função particular (você escolhe) em GBPV2. Por exemplo, a função de “grade\_get\_category\_weight ()” ou “grade\_get\_grade\_items ()” no grau / / lib.php

3. [Teste de Unidade Moodle] Escreva um trecho de código simples usando o Moodle “testes de unidade” quadro para testar uma função particular (você escolhe) em GBPV2. Por exemplo, a função de grade\_get\_category\_weight () ou grade\_get\_grade\_items () in / grade / lib.php.

5. Partilhar os seus pensamentos sobre os temas de discussão para esta Unidade.

Unidade 9: Documentos de Requisitos

## Visão geral | Unidade 9 “fazer” lista

Recursos externos abrem em novas janelas do navegador.

Documentos de requisitos são a cola que mantém a comunicação entre um programador e um usuário juntos. Nesta unidade, vamos explicar como desenvolver um documento de requisitos que vão ajudar a solidificar o motivo, um projeto deve ser concluído, reduzir creep recurso, auxiliar na garantia de qualidade, bem como identificar o que a funcionalidade de um programador deveria estar trabalhando para criar.

Nesta unidade, vamos estar se referindo à estrutura de um documento de requisitos, como descrito no arquivo PDF, “Escrever um Documento de Requisitos de Software”. Por favor, use isso como uma referência e leia atentamente. Leia também através de documento da NASA, Redação Eficaz Especificações de Requisitos, para ter uma idéia de como a palavra de um documento de requisitos para a melhor legibilidade possível.

Por que escrever uma documento de requisitos?

“Os produtos com as contagens mais baixas defeito também têm o menor horários.” M. Tim Jones no controle de versão para Linux: Visão de arquiteturas, modelos e exemplos.

Um documento de requisitos:

- \* É um ponto de partida para garantir que você fizer isso direito o que significa que entregar o que o cliente quer a um nível de qualidade que é aceitável.
- \* Pode ser usado como um “contrato” solta definir qual funcionalidade será entregue.
- \* É escrito como todos que os envolvidos podem entender o documento, incluindo clientes, gerente de projeto, desenvolvedores e testadores.
- \* Garante ao cliente que a equipe de desenvolvimento compreende o que é que eles querem.
- \* Lista claramente as características que devem estar presentes no projeto acabado. Por exemplo, é um documento escrito de que o software fará.
- \* Oferece um ponto de partida para testar a funcionalidade.
- \* É um ponto de partida para a criação de documentação do usuário.
- \* Settles eventuais litígios entre o cliente ea equipe de desenvolvimento
- \* Will lista nada mais, nada menos do que o exigido.
- \* Serve como um ponto de referência para fazer um recurso de congelamento. Ou seja, se as mudanças funcionalidade solicitada, então um documento de requisitos novo é criado eo cliente devem participar neste processo para entender o impacto de fazer mudanças no final de um projeto do ciclo de vida. Isso também pode servir como ponto de partida para a entrega de funcionalidade em versões.
- \* Não se concentrar em como o software irá realizar as tarefas necessárias. Isso é muitas vezes melhor deixar para o programador que será o mais experiente e capaz de tomar decisões de implementação.
- \* Deve ser adaptado para a aplicação. Se o pedido for grande ou pode ser facilmente mal entendido ou mal interpretado, em seguida, o documento de requisitos provavelmente precisam ser mais detalhadas. No entanto, você deve escrever de modo que seja facilmente compreendida.

topo

A entrevista do utilizador

Antes de você pode escrever um documento de requisitos você deve primeiro determinar os requisitos. Uma das melhores maneiras de fazer isso é conversando com um usuário ou grupo de usuários, sejam estes usuários são professores, administradores ou alunos depende do tipo de projeto. Entrevistando o rosto do usuário a cara ou pelo telefone permite que o programador, gerente de projeto e usuário para fazer o seguinte:

1. Chegar a um entendimento comum dos requisitos do projeto
2. Discutir as decisões importantes sobre as características do projeto

3. Discutir as decisões importantes sobre a interface do usuário para o projeto
4. Chegar a um entendimento comum de quanto tempo um projeto levará para ser concluído
5. Identificar quais características são as mais importantes

Abaixo está um conjunto de perguntas que devem ser úteis como ponto de partida para a maioria das entrevistas um desenvolvedor terá com um usuário ou grupo de usuários. Sinta-se livre para modificar isso com questões mais específicas para o projeto, ou adicionar perguntas durante a entrevista como eles vêm à mente. O primeiro conjunto de perguntas é mais geral que o grupo deve responder, eo segundo conjunto é específico e deve ser solicitado de cada usuário do projeto ou programa.

Questões gerais e introdutórias para o grupo inteiro de usuários

1. O que isso vai ajudar a atingir projeto para a organização?
2. Por que estamos fazendo o projeto agora?
3. O que acontecerá se não fizermos o projeto em tudo?
4. Quem se beneficiará com o projeto?
5. As pessoas que se beneficiarão do projeto consideram a melhoria mais importante que pode ser feito neste momento?
6. Devemos fazer um projeto diferente em vez disso?
7. Quando é que o projeto precisa ser concluído?
8. Quais são as partes mais importantes do projeto?
9. Quais são as características do projeto pode ser adiado ou corte se a deslizamentos timeline?

Questões específicas do usuário

1. Qual é o processo atual de executar?
2. Que parte (s) do processo que você gostaria de melhorar?
3. Que serão afetados pelas mudanças que você gostaria?
4. Existem outros sistemas que essa mudança terá de interagir com?
5. Como é que o usuário veja sua interação com o processo atual?
6. Como é que o usuário veja sua interação com o novo processo?
7. Quais as opções que o usuário gostaria de ser capaz de configurar?
8. Que termos não o usuário a entender?
9. Existe alguma coisa que o usuário gostaria que acontecesse quando o projeto está pronto para ser implementado?
10. Quem deve ser capaz de executar as ações dentro do sistema em relação ao novo projeto?
11. Estão lá todos os relatórios que o usuário precisa?
12. Que tipo de requisitos de auditoria que o usuário tem?

topo

A Entrevista Amostra

Idéia do projeto

Corpo docente da Humboldt State University gostaria de ter uma forma de imprimir e registro de frequência de Moodle. Os assistentes de departamento gostaria de reduzir o trabalho de criação de folhas de papel para a faculdade. Equipe técnica da Humboldt State University estão na fase de exploração do desenvolvimento do projeto para determinar a melhor maneira de satisfazer as necessidades dos professores e assistentes de departamento.

Questões gerais e introdutórias para o grupo inteiro de usuários

1. O que isso vai ajudar a atingir projeto para a organização? Permitirá faculdade para ser auto-suficiente quando se trata de tomar atendimento em vez de confiar no departamento de assistência para criar folhas de papel.
2. Por que estamos fazendo o projeto agora? Estamos fazendo esse projeto porque quinze professores e 22 assistentes de departamento têm solicitado esta alteração como sua prioridade máxima nos últimos seis meses.

3. O que acontecerá se não fizermos o projeto em tudo? As coisas vão continuar como estão agora.
4. Quem se beneficiará com o projeto? Assistentes e docentes do departamento.
5. As pessoas que se beneficiarão do projeto consideram a melhoria mais importante que pode ser feita neste momento? Sim, absolutamente.
6. Devemos fazer um projeto diferente em vez disso? Não.

#### Questões específicas do corpo docente

1. Qual é o processo atual de executar? Pedido de folha de presença do assistente de departamento
2. Que parte (s) do processo que você gostaria de melhorar? Gostaria de ser capaz de imprimir folhas de presença 10 minutos antes da palestra em vez de ter de pedir a folha de papel uma semana de antecedência.
3. Que serão afetados pelas mudanças que você gostaria? Assistentes de departamento e professores.
4. Existem outros sistemas que essa mudança terá de interagir com? Não.
5. Como é que o usuário veja sua interação com o processo atual? Vejo minhas interações como restrito. Todas as informações que eu quero é no Moodle, mas eu sou incapaz de acessá-lo como eu quero sem passar por um assistente de departamento.
6. Como é que o usuário veja sua interação com o novo processo? Eu me vejo como menos confinado e capaz de fazer meu trabalho melhor e de forma mais oportuna.
7. Quais as opções que o usuário gostaria de ser capaz de configurar?
  - \* Eu gostaria de ser capaz de configurar quais participantes do curso estão no plantel, no caso eu tenho pessoas que eu não preciso de tomar para atendimento.
  - \* Eu gostaria de configurar a data da folha de papel é para que eu possa imprimir uma diferente.
  - \* Eu gostaria de ser capaz de imprimir a folha de papel para cada vez que minha classe se reúne por apenas entrando nas datas a minha turma se reúne.
8. Que termos não o usuário a entender? Desconhecido neste momento.
9. Existe alguma coisa que o usuário gostaria que acontecesse quando o projeto está pronto para ser implementado?
  - \* Gostaria de ser notificado de que este projeto foi concluído
  - \* Eu gostaria de começar algum treinamento sobre como usar este programa corretamente.
10. Quem deve ser capaz de executar as ações dentro do sistema em relação ao novo projeto? Somente a faculdade deve ser capaz de imprimir a folha de papel.
11. Estão lá todos os relatórios que o usuário precisa? Se eu puder entrar atendimento gostaria de ser capaz de ver um relatório de que tenha frequentado o classes.
12. Que tipo de requisitos de auditoria que o usuário tem? Nenhum.

#### Perguntas assistente do departamento específico

1. Qual é o processo atual de executar?
  1. Logar no Moodle e localizar curso do professor.
  2. Copiar todos os estudantes em uma página do bloco de Pessoas e colá-los em uma planilha.
  3. Repita o passo b para cada página no bloco de Pessoas
  4. Adicionar um campo para a data para a planilha
  5. Adicione um título para o curso para a planilha
  6. Salve o arquivo
  7. Imprimir a planilha
  8. Coloque a folha de rolar na casa do professor
  9. Repita os passos a-h para todas as solicitações
2. Que parte (s) do processo que você gostaria de melhorar? Eu gostaria de remover isso como minha função de trabalho.
3. Que serão afetados pelas mudanças que você gostaria? Assistentes e docentes do departamento.
4. Existem outros sistemas que essa mudança terá de interagir com? Não.
5. Como é que o usuário veja sua interação com o processo atual? Eu tenho que fazer um monte de tarefas repetitivas, copiar e colar. Eu também tenho a impressão e notificar a faculdade, quando suas

folhas de rolo está acabado. Eu provavelmente realizar cerca de 10 horas de trabalho para cada faculdade.

6. Como é que o usuário veja sua interação com o novo processo? Eu não me vejo interagindo com o novo processo.

7. Que termos não o usuário a entender? Desconhecido neste momento

8. Existe alguma coisa que o usuário gostaria que acontecesse quando o projeto está pronto para ser implementado? Gostaria de ser notificado quando o projeto estiver completo e gostaria de saber quem enviar professores para obter treinados sobre como usar o bloco.

topo

## Requisitos Estrutura do Documento

Um documento de requisitos de software não deve mergulhar em detalhes de implementação. Ou seja, o documento deve se concentrar no que o aplicativo deve fazer e não como ele realiza essas tarefas. Este prevê flexibilidade na implementação e ainda entregar as expectativas sólida do que a funcionalidade do programa irá proporcionar.

1. Visão geral da aplicação. O Resumo de aplicações deve ser concluída como um esforço de colaboração entre o gerente de usuário programador, e projeto.

### 1. Objetivos e Justificativas

1. Explique por que este projeto está sendo desenvolvido.

2. Explicar a necessidade do programa ou projeto.

3. Explicar por que isso está sendo feito agora ou mais tarde.

### 2. Processos de Negócio

1. Explicar os processos de negócios que geram a necessidade para esta aplicação.

2. Explicar como o processo de negócio existentes vai mudar.

3. Explique quaisquer outros sistemas ou processos que esta aplicação será dependente.

### 3. Funções de usuário e responsabilidades

1. Descrever as responsabilidades únicas e interações de cada usuário / papel em Moodle terá que executar quando o projeto estiver completo. Por exemplo, um administrador do site terá que configurar as definições ou um estudante é capaz de fazer certas coisas.

2. Para a maior parte das funções gerais que precisam ser definidos são de estudante, professor da faculdade / e administrador.

### 4. Interações com outros sistemas

1. Explicar como o sistema irá interagir com outros sistemas. Por exemplo Moodle irá gerar um arquivo excel para um sistema de estudante a utilizar para carregar notas

2. Definir as interações com todas e quaisquer sistemas externos.

### 5. Substituição de sistemas legados

1. Explicar todos os sistemas de legado que este vai substituir

### 6. Rollout Considerações sobre a produção

1. Explique quaisquer considerações que precisam ser levados em conta quando esta aplicação vai para produção.

### 7. Terminologia

1. Definir qualquer terminologia que um usuário pode não entender completamente

2. Definir todas as siglas utilizadas no documento de requisitos ou em comunicações posteriores sobre o projeto entre os usuários e desenvolvedores.

topo

2. Requisitos funcionais. Requisitos funcionais pode ser preenchido pelo gerente de projeto e programador, com o auxílio do usuário, mas deve sempre ser informado pelo Resumo de aplicações. Esta área do documento de requisitos é mais importante para o programador, porque os detalhes todas as informações que são específicas para execução do programa.

### 1. Declaração de funcionalidade

1. Claramente a funcionalidade necessária do programa. Tenha o cuidado de indicar exatamente o

que é necessário: nem mais nem menos. Este será um contrato livremente ligação entre o cliente ea equipe de desenvolvimento.

2. Em geral para o desenvolvimento Moodle você vai querer quebrar os requisitos funcionais baseados em seus papéis. Se o seu programa é mais uma característica fundamental de todo o sistema, então ele pode ser melhor para a lista de requisitos com base na função.

2. Considerações de segurança

1. Explicar os problemas de segurança que podem precisar de ser tratada como parte do seu aplicativo.

2. Explicar as ações que usuários específicos serão autorizados a fazer ou o que especificamente ações que não podem fazer.

3. Questões de personalização

1. Lista de quaisquer considerações de personalização específicas que podem precisar de ser tratada como parte da implementação do aplicativo. Abordar especificamente o personalizações (se houver) o usuário terá de realizar.

4. Relatórios

1. Descrever todos os relatórios, extração de madeira ou visão geral do sistema que o aplicativo terá que fornecer.

5. Requisitos de Desempenho

1. Lista de todos os requisitos específicos de desempenho aqui. Deve haver métricas específicas e valores aceitáveis na lista, não afirmações vagas ou ambíguas.

6. Usabilidade

1. Lista de quaisquer considerações específicas ou requisitos de usabilidade aqui.

7. Escopo

1. Lista de cada um dos requisitos constantes da Declaração de exigências da função e em que fase do projeto devem ser concluídas até.

2. Geralmente utilizam uma abordagem gradual para listar cada conjunto de requisitos funcionais que serão usados.

3. Nota quando um recurso ou fase é opcional.

3. Suplementares / Background Informação / Apêndices. Lista de todos os documentos complementares, referências ou antecedentes que possam ser necessárias para compreender, implementar, testar, ou usar o aplicativo.

topo

“To Do” Lista

1.

Humboldt State University tem exigências de alguns documentos de amostra em seu sistema Trac. Dê uma olhada nestes dois exemplos: um projeto para permitir crosslisting de cursos e um projeto para agilizar a atualização datas atividade. Também rever este modelo de exemplo para um documento de requisitos.

2.

Leia Tanya Berezin “Escrever um Documento de Requisitos de Software” (documento PDF – Adobe Reader). Qual é a diferença entre o esquema acima e a estrutura do documento requisitos propostos por Tanya Berezin em “Escrever um Documento de Requisitos de Software”?

3. Completar os requisitos redação do documento.

Com base na entrevista de exemplo fornecido na lição desta Unidade, escrever secção 1.1 Objetivos e Justificativas e 1,2 de Processos de Negócios do documento de requisitos. Submetê-lo na área de atribuição designada na seção desta unidade. Certifique-se que o seu documento explica cada um dos pontos importantes listados na estrutura de documentos de requisitos.

Com base na entrevista de exemplo fornecido na lição desta Unidade, escrever secção 2.1 Declaração de Funcionalidade do documento de requisitos. Submetê-lo na área de atribuição



designada na seção desta unidade. Certifique-se que o seu documento explica cada um dos pontos importantes listados na estrutura de documentos de requisitos.

Pensar em uma idéia de projeto que você gostaria de ver implementada no Moodle. Para começar, considere as seguintes idéias:

Projetos fácil

- \* Um bloco que apresenta uma transmissão ao vivo a partir de uma web cam.
- \* Um bloco que armazena a data, hora e local de quando o curso atende

Projeto de dificuldade média

- \* Um bloco que serve como uma calculadora on-line com gráficos, moeda e funções de contabilidade

Projetos difícil

- \* Um bloco de pesquisa do Google Scholar para publicações
- \* Um bloco para pesquisar Amazon para livros
- \* Um bloco que interage com o Google Talk para bater papo enquanto no Moodle
- \* Um bloco que agrega todos os alunos em um curso como amigos no Facebook
- \* Um bloco que agrega todos os alunos em um curso como amigos no MySpace
- \* Um bloco que permite o acesso a um servidor FTP
- \* Um bloco que adiciona o acesso SSH a um servidor
- \* Um bloco que apresenta o weathr locais do Google ou weather.com

Usando o formato de entrevista amostra fornecida nesta Unidade, role-play o processo de entrevista com você mesmo. Apresentar as respostas para as perguntas da entrevista e quaisquer outras perguntas que você sente necessidade de ser convidado para a área de atribuição. Certifique-se de realizar uma entrevista a partir de perspectivas diferentes usuários, tais como o corpo docente e perspectiva do estudante.

Utilizando as respostas para as perguntas da entrevista criar um documento de requisitos completo. Apresentar o documento de requisitos na área de atribuição designada. Certifique-se de explicar todas as perguntas feitas em cada uma das seções da estrutura requisitos acima, bem como fornecer uma captura de tela de um ridicularizado até interface do usuário em um apêndice.

4.

Depois de completar as leituras indicadas e Cessão, escolher entre os seguintes temas de discussão e postar suas idéias.

- \* Comparar e contrastar as diferenças entre o documento HSU de requisitos e do modelo de documento de requisitos descritos na lição desta Unidade.
- \* Qual é a diferença entre o esboço desta lição e da estrutura do documento requisitos propostos por Tanya Berezin em “Escrever um Documento de Requisitos de Software”?
- \* Quais perguntas da entrevista que você adicionar ao conjunto de perguntas da entrevista padrão?
- \* Quais as seções do documento de requisitos que você mudaria? Ou quais seções você acrescentar?

5. Faça o teste. Duas tentativas são permitidas.

## Unidade 10 – Programação de um Bloco Moodle

Visão geral | Unidade 10 “fazer” lista

Recursos externos abertos em novas janelas do navegador.

Ao longo dos últimos nove unidades que evoluíram através dos passos para criar um bloco Moodle, incluindo

- \* Verificar o código de um repositório de código fonte
- \* Aprender sobre algumas ferramentas de desenvolvimento úteis
- \* Criação de Moodle no seu computador
- \* Explorar bibliotecas Moodle útil

- \* Rever informações gerais sobre boas práticas de codificação
- \* Codificação de um bloco em Moodle
- \* Criar casos de teste e execução de scripts Selênio
- \* Escrever um documento de requisitos

Nesta unidade você vai usar tudo o que você aprendeu até agora para criar um bloco de um documento de requisitos, escrever casos de teste, participar de uma revisão de código e, finalmente, liberar o bloco para a comunidade Moodle!

Ao selecionar projetos para o desenvolvimento de blocos neste curso, vários fatores devem ser cuidadosamente considerados. Em alguns casos, os documentos de requisitos desenvolvidos na Unidade 9 não pode produzir a melhor oportunidade de sucesso para programadores iniciantes Moodle, ou possam vir a encontrar resistência significativa quando proposto para a comunidade Moodle, por razões que poderiam ser difíceis de prever.

Procurar o conselho do facilitador do curso sobre a possibilidade de prosseguir com seu projeto para o desenvolvimento Unidade 9 bloco completo. A Unidade 10 Documento de Requisitos fornecido abaixo destina-se a oferecer uma tarefa gerenciável para a conclusão bem sucedida deste curso e deve ser usado para a unidade 10 a menos que o incentivo específico é fornecido pelo facilitador para continuar com sua idéia de projeto Unidade 9.

Você pode trabalhar em equipes de projeto, se desejar. No entanto, a menos que todos os membros da equipe estão na mesma instituição, compartilhando o CVS mesmo, pode haver dificuldades técnicas na gestão de edições para o código. Alertar o seu facilitador dos membros de sua equipe. Ao apresentar seu projeto, você deve documentar as contribuições de cada membro da equipe. Desenvolvimento pós-Documentação

Depois de completar o desenvolvimento de um bloco, existem dois tipos de informação que a comunidade de usuários Moodle precisa para usar o seu bloco:

1. Instalação / Readme documentação
2. Ajuda documentação sobre como usar o bloco

#### Instalação / Readme Documentação

Documentação de instalação informa o administrador do sistema das etapas necessárias para instalar o bloco. Este arquivo, geralmente INSTALL.txt título, normalmente contém instruções sobre como colocar o bloco no diretório blocos e depois fazer login no Moodle como administrador e clicar em “Notificações” link no bloco de Administração do Site. Se, no entanto, o bloco é mais complexo do que um bloco simples, como um cron, ou o bloco modifica os arquivos do núcleo do Moodle código (não recomendado), então o arquivo de instalação terá de conter informações sobre como esses arquivos serão modificados.

O arquivo readme, geralmente chamado README.txt, contém documentação sobre quem desenvolveu o bloco, como contatá-lo com perguntas e qualquer outra informação que é informativa para o usuário e fala sobre si mesmo e do projeto.

Um arquivo final opcional é o arquivo Changelog, que se destina a informar o utilizador sobre as mudanças que foram feitas em um bloco em cada versão. Este arquivo é opcional no primeiro lançamento, mas é recomendado em cada liberação subsequente. Este arquivo deve conter todos os bugs que foram corrigidos na versão atual, possivelmente apontando para os números de bilhete específico em um sistema de rastreamento do bilhete.

Dica importante: tanto o README e INSTALL arquivos devem ser salvos em um formato comum em todas as plataformas. Arquivos de texto (. Txt e. Rtf) são recomendados.

#### Documentação de Ajuda

A documentação final exigida é a documentação de Ajuda. Isto vem na forma de / e como a arquivos, wikis, FAQs, ou vídeo. Isto deve ensinar ao usuário como configurar e usar o bloco.

Recomenda-se que você tem tanto uma faculdade e guia do estudante para a maioria dos blocos. Este também vem em forma de arquivos HTML no Moodle. Esses arquivos estão associados com o azul de ponto de interrogação botões em algumas formas. Este botão é impresso usando o `HelpButton ()` em `weblib.php`

topo

“To Do” Lista

Planejamento do Desenvolvimento Bloco

1.

Escolha um projeto de sua escolha ou escolha da lista fornecida na Unidade 9 do “fazer” a lista de itens, # 3 para criar um documento de requisitos para este projeto, conforme descrito na Unidade 9.

2.

Leia os detalhes fornecidos em Moodle.org, Trabalhando com a Comunidade, para se familiarizar com o processo de partilha de código com a comunidade Moodle. Preste atenção especial para a seção de perto da parte inferior da página, “Considere pedir ou discutir sua proposta em primeiro lugar.”

3.

Compartilhar informações com a comunidade Moodle sobre o que você pretende desenvolver na área relevante fórum de discussão em Moodle.org.

4. Documento testes de unidade que são necessários para testar o bloco com base no documento de requisitos.

Criação do Bloco

5.

Leia este recurso geral, Escrita de Documentação Software Bom, e considerar como ele se aplica ao seu projeto de bloco Moodle.

6.

Escreva o código de bloco com base no documento de requisitos.

7.

Escrever casos de teste testando cada um dos requisitos do documento de requisitos.

8.

Escrever casos de teste para testar a funcionalidade Selenium seu bloco.

9.

Compor a sua documentação em desenvolvimento, bem como pós-desenvolvimento documentação.

10.

Anexar o seu código no Fórum designados neste curso como um anexo a uma mensagem no fórum (zip seu diretório primeiro bloco). Participar de uma revisão de código de código pelo menos dois outros colegas “, com o facilitador. Publique o seu feedback e comentários de revisão feitas em seu código. Considerar que os comentários devem ser incorporadas para melhorar o seu código e fazer as alterações.

11.

Depois de fazer as edições finais ao seu código, apresentar seu diretório bloco final zipado na área de atribuição designada. Se esqueça de incluir seus arquivos pós-desenvolvimento de documentação e casos de teste dentro de seu arquivo compactado.

topo

Contribuir com o Código para a Comunidade Moodle

Agora que o bloco está completo eo diretório zipado bloco é lançado para o Assignment estão neste

curso, o próximo passo é liberar o bloco para a Comunidade Moodle. Siga os passos abaixo 12/09 (e opcionalmente as etapas 13 e 14) para liberar seu código para a comunidade Moodle.

Apenas a um quarto do projeto será lançado para a comunidade. Use a escolha na seção final dos materiais deste curso é para votar para o melhor bloco que deve ser compartilhado. Considerar os seguintes critérios em seu voto:

- \* Manutenção de código
- \* Consistência com as diretrizes de codificação Moodle
- \* Acessibilidade
- \* Usabilidade
- \* Os requisitos que satisfaçam as especificações do documento
- \* A qualidade de casos de teste
- \* Documentação

O autor (s) do bloco “vencedora” será atribuído ao concluir as etapas 9-12 abaixo (e opcionalmente as etapas 13 e 14) para completar a liberação.

9.

Postar o arquivo zip para um servidor publicamente acessível para permitir que outros para baixar o arquivo.

10.

Criar uma nova entrada para o bloco no “Módulos e Plug-ins” banco de dados

- \* Use um nome descritivo para o bloco. Selecione um nome descritivo e único para identificá-lo.
- \* Descreva o que o bloco se destina a fazer
- \* Dê uma captura de tela do bloco em ação no seu local de testes locais e pós a captura de tela na área de captura de tela
- \* Link para baixar o Moodle 1.8 para o servidor onde você carregou o arquivo
- \* Digite seu nome como o mantenedor

11.

Criar um novo tópico no fórum para o bloco no fórum de discussão Blocks

(<http://moodle.org/mod/forum/view.php?id=2121>) com as seguintes informações

- \* O que o projeto faz
- \* Contribuintes
- \* Documento Requisitos
- \* Os casos de teste
- \* Os casos Selenium
- \* Link para o módulo de entrada e plugins

12.

O link “Módulos e Plugins” entrada para o tópico fóruns.

13.

Opcional: Adicione seu projeto para o Tracker Bug Moodle

- \* Acesso ao [tracker.moodle.org](http://tracker.moodle.org)
- \* Clique no botão “Componentes” link no âmbito do projecto “não-core contribuiu módulo” no lado inferior esquerdo
- \* Clique em “Adicionar um projeto aqui”
- \* Clique em “Criar nova edição”
- \* Insira as informações para o ticket.

14.

Opcional: Pedido de um diretório contrib criado para o projeto do ramo Moodle contrib CVS.

- \* Acesso ao [tracker.moodle.org](http://tracker.moodle.org)
- \* Clique no botão “Componentes” link no âmbito do projecto “não-core contribuiu módulo” no lado

inferior esquerdo

- \* Clique em “Adicionar um projeto aqui”

- \* Clique em “Criar nova edição”

- \* Insira as informações para o bilhete de problema com o título e um texto solicitando uma pasta no diretório contrib bloco ser feitas para o seu projeto.

15.

Participar na discussão sobre o gerenciamento de código atribuído em uma comunidade distribuída.